## Contents

**Preface** ......................................................... 7  
Documentation Conventions ........................................ 9  
Documentation Set .................................................. 11

**What’s New in Version 520** .................................. 13  
Descriptions of the New Features ................................. 14

**Chapter 1. Getting Started in Omegamon II for MVS** ........ 17  
Signing On and Off OMEGAMON II .................................. 18  
Understanding the User Interface .................................. 21  
Moving about Quickly in OMEGAMON II ............................. 33  
Obtaining Online Help ............................................... 38

**Chapter 2. Adjusting Your OMEGAMON II Environment** ..... 47  
Controlling the Appearance and Content of Panels .............. 48  
Adjusting Thresholds ................................................. 53  
Understanding OMEGAMON II Profiles .............................. 56  
Setting OMEGAMON Logging Activity On or Off ................. 60  
Synchronizing OMEGAMON II to Local System Time ............. 61

**Section 1.**  
**IPS-Based Performance Monitoring**

**Chapter 3. Monitoring System Performance** ................ 65  
Interpreting Status Lights ........................................... 66  
Monitoring Workloads, Resources, and Alerts ..................... 71  
Monitoring Performance across Systems ........................... 129  
Monitoring Historical Shared-DASD Performance ................. 134

**Chapter 4. Solving Workload Delays** .......................... 137  
Understanding Bottleneck and Impact Analysis .................... 138  
Collecting Bottleneck and Impact Analysis Information ........ 140  
Excluding Wait Reasons from Bottleneck Analysis ............... 143  
Example: Diagnosing a Current Bottleneck ........................ 145  
Example: Analyzing a Recurring Bottleneck ....................... 150  
Guidelines for Troubleshooting Bottlenecks ....................... 154

**Chapter 5. Developing Your Own Realtime Screens** .......... 157  
Creating/Modifying Your Screen Space ............................ 158  
Steps in Creating or Modifying a Screen Space .................... 159  
Guidelines for Creating Your Own Screen Spaces ................. 161
Chapter 6. Gaining Historical Perspective

Chapter 7. Designing Your Own Historical Reports

Section 2. WLM-Based Performance Monitoring

Chapter 8. Monitoring System Performance

Chapter 9. Solving Workload Delays

Chapter 10. Gaining Historical Perspective

Overview
| Chapter 11 | Designing Your Own Historical Reports | 347 |
| Creating a Historical Output File | 348 |
| Producing SAS Graphical Reports | 359 |
| Appendix A | Running OMEGAMON II from a Dedicated (Non-VTAM) Terminal | 373 |
| Signing Onto and Off OMEGAMON II | 374 |
| Adding and Releasing Sessions | 375 |
| Changing the Polling Interval | 376 |
| Setting up and Using Automatic Sign On | 377 |
| Appendix B | Guide to Candle Customer Support | 379 |
| Base Maintenance Plan | 380 |
| Enhanced Support Services | 384 |
| Customer Support Contact Information | 385 |
| Index | 387 |
About this document

This guide is an introductory tool to help operators, data center personnel, systems programmers, performance analysts, and data center managers monitor and tune their MVS systems using OMEGAMON II® for MVS.

This guide contains the following information:

- an overview of OMEGAMON II features, the types of panels displayed and how to navigate from one panel to another
- instructions for adjusting your OMEGAMON II environment
- usage scenarios describing how to use OMEGAMON II to monitor the realtime and historical performance of your system

About MVS 5.2

MVS 5.2 introduces MVS users to Workload Manager (WLM), an intuitive business-oriented way to manage work and resources on your computer system. Under MVS 5.2, you will be using an operating system that can run in either of two modes: compatibility mode or goal mode.

When MVS 5.2 is running in compatibility mode, workloads and resources are managed by parameters that you specify in System Resource Manager (SRM) IEAIPSxx and IEAICSxx parmlib members.

When MVS is running in goal mode, workloads and resources are managed by WLM according to performance goals that you specify in the WLM service policy. WLM handles the tradeoffs between meeting performance goals and using system resources efficiently.
About MVS 5.2 and OMEGAMON III

OMEGAMON II responds as follows when MVS/SP 5 switches from compatibility mode to goal mode or vice versa:

- OMEGAMON II dynamically adjusts without user intervention and displays a pop-up window announcing the switch to the new mode.
- The OMEGAMON II user interface operates the same in the new mode.
- The OMEGAMON II panels display different objects and different performance-monitoring data in the new mode.

About MVS 5.2 and this document

This guide supports OMEGAMON II running in compatibility and goal modes as follows:

- Chapter 1 describes how to operate the user interface.
- Chapter 2 describes how to adjust your OMEGAMON II environment, including panel display options and performance monitoring threshold settings.
- Chapters 3 through 7 (Part I) describe how to use OMEGAMON II to monitor your system when MVS is running in compatibility mode.
- Chapters 8 through 11 (Part II) describe how to use OMEGAMON II to monitor your system when MVS is running in goal mode.
Documentation Conventions

Introduction
Candle documentation adheres to accepted typographical conventions for command syntax. Conventions specific to Candle documentation are discussed in the following sections.

Panels and figures
The panels and figures in this document are representations. Actual product panels may differ.

Revision bars
Revision bars (|) may appear in the left margin to identify new or updated material.

Variables and literals
In examples of command syntax, uppercase letters are actual values (literals) that the user should type; lowercase letters are used for variables that represent data supplied by the user. Default values are underscored.

LOGON APPLID (cccccccc)
In the above example, you type LOGON APPLID followed by an application identifier (represented by cccccccc) within parentheses.

In ordinary text, variable names appear in italics.
Symbols

The following symbols may appear in command syntax:

Table 1. Symbols in Command Syntax

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Usage</th>
</tr>
</thead>
</table>
| | The “or” symbol is used to denote a choice. Either the argument on the left or the argument on the right may be used. Example:  
|**YES | NO**  
In this example, YES or NO may be specified. |
| [ ] | Denotes optional arguments. Those arguments not enclosed in square brackets are required. Example:  
|**APPLDEST DEST [ALTDEST]**  
In this example, DEST is a required argument and ALTDEST is optional. |
| { } | Some documents use braces to denote required arguments, or to group arguments for clarity. Example:  
|**COMPARE {workload} -**  
**REPORT={SUMMARY | HISTOGRAM}**  
The **workload** variable is required. The REPORT keyword must be specified with a value of SUMMARY or HISTOGRAM. |
| _ | Default values are underscored. Example:  
|**COPY infile outfile - [COMPRESS={YES | NO}]**  
In this example, the COMPRESS keyword is optional. If specified, the only valid values are YES or NO. If omitted, the default is YES. |
Documentation Set

Introduction

Candle provides a complete set of documentation for OMEGAMON II for MVS. Each manual in this documentation set contains a specific type of information to help you use the product.

- The *OMEGAMON II for MVS Configuration and Customization Guide* provides instructions for configuring and customizing OMEGAMON II for MVS after it is installed.
- The *OMEGAMON II for MVS User’s Guide* describes how to use OMEGAMON II for MVS panels to monitor your system’s performance, solve performance problems, and prevent problems from recurring.
- The *OMEGAMON II for MVS Messages Manual* lists messages you may receive while running OMEGAMON II for MVS and explains what to do when they appear.
- The *OMEGAMON II for MVS Command Language Reference Manual* provides comprehensive descriptions of the OMEGAMON II for MVS commands and keywords, arranged in alphabetical order.
- The *Epilog for MVS Command Language Reference Manual* provides comprehensive descriptions of the Epilog for MVS commands and keywords, arranged in alphabetical order.
- The *OMEGAMON II for MVS Quick Reference* explains how to use the CUA interface and lists all of the fast paths. This book is available in pocket-size format.

We would like to hear from you

Candle welcomes your comments and suggestions for changes or additions to the documentation set. A user comment form, located at the back of each manual, provides simple instructions for communicating with the Candle Information Development department.

You can also send email to UserDoc@candle.com. Please include “OMEGAMON II for MVS User’s Guide V520” in the subject line.
What’s New in Version 520

What’s new

This chapter provides information about the new features provided with OMEGAMON II for MVS, Version 520.

Contents

- New fields on the Set Service Class Thresholds panel . . . . 14
- Support for the IBM 2105 Enterprise Storage Server . . . . 14
- Bottleneck analysis for multi-tasking jobs . . . . . . . . . . . 14
- Service class information now available through a zoom to Epilog session . . . . 14
- Threshold for CPs expected online . . . . . . . . . . . . . . . . . . . . . . 15
- Online documentation provided in PDF format . . . . . . . . . . . . 15
Descriptions of the New Features

New fields on the Set Service Class Thresholds panel

New fields have been added to the Set Service Class Thresholds panel to permit you to qualify goal reporting for a specific service class. The qualification applies to either Velocity goals or Response Time Goals.

On this panel, if the Goal Type is specified as Velocity, then an updatable field, with CPU percent busy, is presented to you. If your system is displaying a workload warning or critical light that is unexpected or misleading, you can specify a percentage in this field to identify the minimum percentage of CPU that must be consumed before OMEGAMON II for MVS displays the warning or critical light.

If the Goal Type is specified as Response time, then the updatable field, with transaction rate is presented. Your response is entered in two parts, the number of transactions and a time qualifier, that may be h, m, or s (hours, minutes, or seconds). This field represents the number of transactions per time period that must occur before OMEGAMON II for MVS displays a warning or critical light.

Support for the IBM 2105 Enterprise Storage Server

This release of OMEGAMON II for MVS provides cache memory reports for IBM’s 2105 Enterprise Storage Server. These statistics include cache read and write hit percentages and I/O requests indirectly related to the cache. Both realtime and historical data is available.

Bottleneck analysis for multi-tasking jobs

Some workloads are capable of performing in multiple execution states simultaneously. An example is a workload that is actively using CPU while delayed waiting for a tape mount. Such workloads are considered multi-tasking and, therefore, capable of achieving resource total percentages greater than 100%.

Service class information now available through a zoom to Epilog session

You can now use the Display command to display reports about workload manager data. The Workload Manager Service Classes report will provide information, by service class, on the goal type and importance, its duration, percent of transactions to complete within the specified response time goal, as well as the transaction count, transaction rate, I/O rate, average storage, and CPU percent. Display commands you can use include:

- DIS RSCl displays summary information about service classes from the perspective of the WLM (goal information and performance index). You can also use the display command to view three detail reports:
  - DIS SCL(nnnnnnnn) to display service class information, where nnnnnnnn is a service class name.
  - DIS WKL(nnnnnnnn) to display workloads, where nnnnnnnn is a workload name.
  - DIS RPC(nnnnnnnn) to display report classes, where nnnnnnnn is a report class name.
Threshold for CPs expected online

The **Set CPU Exception Thresholds** panel permits you to specify a value called CPs expected online. If, during a sampling interval, the actual number of central processors (CPs) detected as being online is less than the value you specified, the CPU light on the System Status panel turns red. If the number of CPs online is greater than the value you specified, the light turns yellow.

You can analyze the problem by navigating to the Analyze CPU Problems panel or navigate directly to the System Environment panel for details.

Online documentation provided in PDF format

With this release of OMEGAMON II for MVS, Candle Corporation has moved the manuals from IBM’s BookMaster to Adobe FrameMaker. This move was made to better enable us to address our customers’ needs by providing tools that enhance productivity.

One of the results of the move is that it is no longer possible to create BookManager versions of the OMEGAMON II for MVS manuals. However, the manuals remain available online in the Adobe PDF version on CD-ROM and are also available on the Candle corporation website at www.Candle.com.

The documentation CD provided with this release has robust and easy-to-use search capabilities. You can search for information in multiple volumes, multiple versions, and across products. The CD also provides easy setup of search indexes with a single click of the mouse.

If you want to order printed copies of the documentation, please contact your Candle Support Services representative.
Chapter Overview

This chapter presents the information necessary for you to understand the structure of the OMEGAMON II user interface. You will learn all the parts of the user interface and what you can do in each part. Then you will learn how to move around in the product to accomplish various tasks. Later chapters assume that you understand the user interface, know how to move quickly from one part to another, and know how to access online help.

Compatibility mode vs. goal mode

The OMEGAMON II user interface operates the same way in compatibility and goal modes. You can use the instructions in this chapter to help you operate the user interface at any time, regardless of the mode in which MVS is running.

Panel illustrations in this chapter

We use the compatibility-mode version of panels and pop-ups throughout this chapter to illustrate the operation of the user interface. If MVS is running in goal mode, do not be concerned that the objects on your screen do not match the objects in the illustration in this chapter. The user interface instructions are the same for both modes.

Chapter Contents

- Signing On and Off OMEGAMON II ................................................. 18
- Understanding the User Interface ............................................... 21
- Moving about Quickly in OMEGAMON II ...................................... 33
- Obtaining Online Help ............................................................... 38
Signing On and Off OMEGAMON II

In this section you learn how to sign onto and sign off OMEGAMON II. In this section we cover the following information:

- "Signing Onto OMEGAMON II," below
- "Signing Off OMEGAMON II" on page 19

Signing Onto OMEGAMON II

You access OMEGAMON II by logging onto a VTAMtm application. The first panel that you see is the Candle logo panel, which welcomes you and presents copyright information.

Sign On panel

The Sign On panel controls access to OMEGAMON II and enables you to set certain logon options before proceeding.

To go to the Sign On panel, press Enter from the Candle logo panel. The following panel is displayed.

```
Omegamon II for MVS - V520
OMII-MVS-520

Sign On Panel

Type the requested information, then press Enter.

Identification
  Userid . . . . . . ________
  Password . . . . . ________
  Change Password. . No + Yes/No

Additional Information
  Group . . . . . . ________
  Account . . . . . ________

F1=Help  F3=Exit  F4=Prompt  F6=Panel ID  F11=Logon Options
```

Sign On panel input fields

The Sign On panel contains input fields where you must enter your user ID and password, and any accounting information required. The security system at your site prescribes the information needed on the Sign On panel. Contact your security administrator for further details.

For more information about an input field, move the cursor to the field and press the function key labelled F1 on your keyboard.
Completing the sign on
To complete the sign on to OMEGAMON II, fill in the appropriate fields on the Sign On panel and press Enter.

An interim panel entitled Establishing the OMEGAMON II Environment is displayed, followed by the System Status panel. Work in OMEGAMON II begins from the System Status panel.

Signing Off OMEGAMON II
To sign off OMEGAMON II, you must be on the System Status panel and OMEGAMON II. Follow the procedure below.

- If you are on any panel other than the System Status panel, press F15 to return to the System Status panel, then press F3.
- If you are on the System Status panel, press F3.

Exit confirmation
After you press F3, the Exit Confirmation pop-up window is displayed, asking you to verify that you want to exit OMEGAMON II. The window gives two choices, exit or return to OMEGAMON II, as shown in the following screen.
Exit OMEGAMON II
To exit OMEGAMON II, perform one of the following actions:
- Press Enter, or
- Press F3, or
- Type x or 1 on the input line before Exit and press Enter
Understanding the User Interface

Introduction
In this section you will learn how OMEGAMON II:

- presents information on panels
- displays selections on pull-down menus
- provides additional information on pop-up windows

In this section we cover the following topics:

- “Learning about Panels” on page 21
- “Recognizing Selectable Items” on page 26
- “Learning about Pull-down Menus” on page 26
- “Learning about Pop-up Windows” on page 27
- “Scrolling on the Panels” on page 29
- “Interpreting Information on Panels” on page 30

Learning about Panels
We use the System Status panel to illustrate the OMEGAMON II panel format. The System Status panel is the first panel you see when you sign onto OMEGAMON II, the point from which all actions begin.

The System Status panel, like all other OMEGAMON II panels, consists of three parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Name</th>
<th>Position on the Panel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Action bar</td>
<td>Top line of the panel.</td>
</tr>
<tr>
<td>2</td>
<td>Panel body</td>
<td>Everything between the top line and the bottom line of the panel.</td>
</tr>
<tr>
<td>3</td>
<td>Function keys</td>
<td>Bottom line(s) of the panel.</td>
</tr>
</tbody>
</table>

Typical panel layout
Each of the three parts of the panel is shown in the following figure.
The sections immediately following discuss the parts of the panel.

**Action bar**

The action bar is the top line of the panel. It has an input field on the left side and available selections spread across the length of the line. The following table lists the available selections and when to use them.

<table>
<thead>
<tr>
<th>Selection</th>
<th>When to Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions</td>
<td>To perform specific actions upon objects in the panel body.</td>
</tr>
<tr>
<td>GoTo</td>
<td>To access additional related panels.</td>
</tr>
<tr>
<td>Index</td>
<td>To navigate quickly to any panel in the product from a hierarchical menu structure arranged by topic.</td>
</tr>
<tr>
<td>View</td>
<td>To filter and/or sort data presented on the panel. Use it to filter out unwanted information and focus only on specific information.</td>
</tr>
<tr>
<td>Options</td>
<td>To set or change the appearance or content of panels.</td>
</tr>
<tr>
<td>Help</td>
<td>To receive online help.</td>
</tr>
</tbody>
</table>
Using the action bar
There are two ways to make selections from the action bar, as shown in the following table.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type the first letter of the action choice in the action bar input field (upper left corner of screen), and press Enter. For example, for Help, type h and press Enter. <strong>Note:</strong> When using Actions from the action bar selections, you must select an object on the panel body first. See the following part for detailed instructions. OR Press the tab key to move the cursor to the action bar choice, and press Enter.</td>
<td>A pull-down menu appears from which you can make a selection.</td>
</tr>
</tbody>
</table>

Panel body
OMEGAMON II displays data in the panel body. The format varies depending on the type of data presented, including status indicators, data lists, and graphic representations.

Acting upon an object in the panel body
An object is defined as any item monitored by OMEGAMON II. Objects are presented on the panel body and are preceded by an input field so you can select and act upon them.

There are three approaches to acting on an object in the panel body, as shown in the table below. For all approaches, use the return key or tab key to place the cursor on the input line before the object chosen. For purposes of this example, we assume you are at the System Status panel and that you are using Batch as the object. Place the cursor on the input line before Batch to continue with the example.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Result</th>
<th>When to use it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type slash (/) on the input line before Batch and press Enter.</td>
<td>The Actions pull-down menu appears from which you can make a selection.</td>
<td>Use this approach if you are not sure where to start your search and want to see the options available.</td>
</tr>
</tbody>
</table>
Type a mnemonic for an action item. For example, type `t` (mnemonic for Historical Trends) on the input line before the object, Batch.

In this example, a pop-up window prompts you for a group number or name. Type in one or both, and press Enter. The next panel asks you to select a date and time for data display. Then historical trends of any batch job currently running are shown.

Use this approach if you want to obtain specific information, such as historical trends, as in this example.

Place the cursor on the Batch input line and press Enter.

A more detailed panel appears about the object, Batch.

Use this approach to show further details about the selected object.

### Function keys

Function keys cause a predefined operation to occur. Many of the tasks you perform with OMEGAMON II use function keys. These keys are labelled as the Fn keys, such as F1, F2, and F3. Function key descriptions are found on the bottom lines of the panel. Each panel lists the available function keys and their assignments for the current panel. If your panel does not display function keys at the bottom, press F2 to turn on the display. (See the F2 function key description in the following table.)

### Function key assignments

Function key assignments are listed below.

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Name</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Help</td>
<td>Displays a help window for the current panel, pull-down menu, pop-up window, or field.</td>
</tr>
<tr>
<td>F2</td>
<td>Keys</td>
<td>Turns on (or off) the display of the function key area at the bottom of the panel. If the function key area is currently displayed, and you press F2, it disappears. If you press F2 again, the assignments reappear.</td>
</tr>
<tr>
<td>F3</td>
<td>Exit</td>
<td>Exits the current panel, saving all changes and additions made in any of the fields. From the System Status panel, you exit the product. From all other panels, you are returned to a previous panel.</td>
</tr>
<tr>
<td>F4</td>
<td>Prompt</td>
<td>Displays a list of valid selections for an input field. Selections denoted with a plus sign (+) can be prompted. If there are only two valid selections, such as on or off, F4 switches between the two.</td>
</tr>
<tr>
<td>F5</td>
<td>Refresh</td>
<td>Updates the panel with current data.</td>
</tr>
<tr>
<td>F6</td>
<td>Console</td>
<td>Takes you to the System Console panel.</td>
</tr>
</tbody>
</table>
Understanding the User Interface

**Note:** F2, F5, F6, and F11 are assigned differently on the Help panels. See “Obtaining Online Help” on page 38 for a description of these keys.

### Navigation for the new user

You navigate around OMEGAMON II using conventional keys - Enter key, tab keys, backspace key, arrow keys, and function keys (when applicable). Using these keys you can locate the object you want on a panel body, then act upon it.

If you are a new user, the following three items of information are all you need to know to navigate when you first begin:

- Move the cursor to the input field before an object in the panel body about which you would like to show details and press Enter.
- Press F1 for help.
- Press F3 to go back to the previous panel. Continue to press F3 to go back to the System Status panel.

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F7</td>
<td>Backward</td>
</tr>
<tr>
<td>F8</td>
<td>Forward</td>
</tr>
<tr>
<td>F9</td>
<td>Function keys help</td>
</tr>
<tr>
<td>F10</td>
<td>Action bar</td>
</tr>
<tr>
<td>F11</td>
<td>Print</td>
</tr>
<tr>
<td>F12</td>
<td>Cancel</td>
</tr>
<tr>
<td>F15</td>
<td>System status</td>
</tr>
<tr>
<td>PA2</td>
<td>Return</td>
</tr>
</tbody>
</table>
Recognizing Selectable Items

OMEGAMON II uses colors, highlighting, underlining, and parentheses to indicate selectable items.

Color terminals

On color terminals, items available for selection appear in white; items not available for selection appear in blue. For example, suppose you select object TSO: RTA, request the Actions pull-down menu, and the Bottleneck action choice is blue. This indicates that the Bottleneck action is not available for this object at this time.

Input fields are denoted by a green line upon which you can type.

The mnemonic letter for a selectable menu item is highlighted. For example, if Historical Details is a valid selection for the current object, the H is highlighted or underlined.

Non-color terminals

On non-color terminals, fields that accept input are underlined.

The mnemonic letter of the selectable menu item appears in parentheses at the end of an item if it is different from the first letter of the selectable menu item in most cases. When this is not true, a message appears, and you must select the number of the desired item.

Learning about Pull-down Menus

A pull-down menu appears after you make a selection from the action bar. This menu provides further selections to help you indicate the exact action you would like to take.

Using a pull-down menu

To use a pull-down menu, follow the steps below.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Type the first letter of an action choice (for example, I for Index) in the action bar input field, and press Enter. OR Use the tab key to go to a choice on the action bar, and press Enter. Exception: For the Actions choice, you must first select an object from the panel body by typing slash (/) on the input line before the object (for example, Batch) and pressing Enter.</td>
<td>The pull-down menu for that selection appears.</td>
</tr>
</tbody>
</table>
Understanding the User Interface

Example of a pull-down menu

The following panel illustrates the GoTo pull-down menu.

Learning about Pop-up Windows

Pop-up windows appear superimposed over the body of the panel in response to selections made on a menu or on the panel body. Pop-up windows may:

- present further selections
- provide alerts for possible error conditions
- display help
- solicit input

More than one pop-up window may be visible at a time, and the windows can overlap. Available function keys are displayed at the bottom of the window.
Example of a pop-up window

The panel below illustrates a pop-up window that is displayed if you request t (Historical Trends) for the Batch object on the System Status panel body. It is a pop-up window that prompts you for a specific group number or name in order to display the desired information in the next panels.

Note: In the previous panel, note that both group number and name have a plus sign (+) after them. The plus sign (+) indicates that you can use F4 to see a list of selections that may be used in the input field.

If you press Enter on the previous panel, another pop-up window appears, as shown below. From the Trend Date/Time Selection window, you can select from a range of options regarding the display of historical trend data. After making a selection on this window, the requested historical trends are displayed.
F12=Cancel appears at the bottom of a pop-up window. Press F12 to close a pop-up window.

---

### Scrolling on the Panels

You can scroll from one screen of data to the next using the function keys or scroll to a specific line of data by specifying a beginning line number.

#### Scrolling function keys

Use the following function keys to scroll.

<table>
<thead>
<tr>
<th>USE...</th>
<th>TO...</th>
</tr>
</thead>
<tbody>
<tr>
<td>F7</td>
<td>Scroll backward one screen at a time.</td>
</tr>
<tr>
<td>F8</td>
<td>Scroll forward one screen at a time.</td>
</tr>
</tbody>
</table>
**Example of scrolling by specifying a beginning line number**

On tabular panels of scrollable data, there is an input field labeled Lines, as shown in the following figure.

You can use the Lines input field to locate the beginning line of a range of 26 lines (one panel). If, for example, you want to go quickly to line 102, you would type 102 in the input space to the right of **Lines**. Note that you can only enter the starting point number. The panel would show lines 102 to 127 of 179.

**Interpreting Information on Panels**

OMEGAMON II presents information in numerical and/or graphical form using the same set of conventions from panel to panel.
### Example of data presentation

In the panel below, CPU % is given numerically and depicted graphically as horizontal lines with a numeric scale.

<table>
<thead>
<tr>
<th>STC Name</th>
<th>Step</th>
<th>Elapsed</th>
<th>Wait</th>
<th>IO Rate</th>
<th>CPU %</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TDGCS03</td>
<td>TDGCS03</td>
<td>21:47 mn</td>
<td>0.0</td>
<td>11.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>TDOCS21</td>
<td>TDOCS21</td>
<td>26:25 mn</td>
<td>0.0</td>
<td>2.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>$OMVTAMG</td>
<td>OMVTAM</td>
<td>08:26 hr</td>
<td>0.0</td>
<td>2.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>GRS</td>
<td>GRS</td>
<td>11:59 hr</td>
<td>0.0</td>
<td>2.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>TDM2S21</td>
<td>TDM2S21</td>
<td>02:30 hr</td>
<td>0.0</td>
<td>1.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>EMFG</td>
<td>EMFG</td>
<td>11:57 hr</td>
<td>0.0</td>
<td>1.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>TDGCS01</td>
<td>TDGCS01</td>
<td>20:41 mn</td>
<td>05:12 mn</td>
<td>0.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>TDOCS02</td>
<td>TDOCS02</td>
<td>18:02 mn</td>
<td>0.0</td>
<td>0.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>TDOCS21</td>
<td>TDOCS21</td>
<td>17:59 mn</td>
<td>5 s</td>
<td>0.0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>TDCSHS03</td>
<td>TDCSHS03</td>
<td>07:20 mn</td>
<td>48 s</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M2G</td>
<td>M2G</td>
<td>11:56 hr</td>
<td>3 s</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDM2S80</td>
<td>TDM2S80</td>
<td>36:51 mn</td>
<td>05:40 mn</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDOCS23</td>
<td>TDOCS23</td>
<td>26:18 mn</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NMFG</td>
<td>NPM</td>
<td>11:54 hr</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NETVSSG</td>
<td>NETVSS</td>
<td>11:53 hr</td>
<td>11:53 hr</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NETVGS</td>
<td>NETVIEW</td>
<td>11:53 hr</td>
<td>1 s</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ST250G</td>
<td>ST250</td>
<td>11:53 hr</td>
<td>27 s</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conventions used when presenting data in graphs

OMEGAMON II uses the following conventions when presenting data.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric scale</td>
<td>x......y</td>
<td>Appears in the header of a graph. The leftmost number (x) indicates the lowest value of the scale and also represents the increment on the scale. The rightmost number (y) indicates the highest value of the scale. Refer to the preceding panel where each dot from the lowest to the highest value of the scale represents one percentage point.</td>
</tr>
<tr>
<td>Horizontal line</td>
<td>--------</td>
<td>Represents the value of the observation measured against the scale.</td>
</tr>
<tr>
<td>Single arrow</td>
<td>&gt;</td>
<td>Represents the end value of the observation measured against the scale. Any value above zero is represented by at least a single arrow.</td>
</tr>
<tr>
<td>Double arrow</td>
<td>&gt;&gt;</td>
<td>Indicates that the value exceeds the upper limit of the scale.</td>
</tr>
<tr>
<td>Exclamation point</td>
<td>!</td>
<td>Indicates a warning or critical threshold. On color terminals, the exclamation point is displayed in the color that represents a warning or critical threshold. To see the value of a threshold, place the cursor on an exclamation point and press F1.</td>
</tr>
</tbody>
</table>
Moving about Quickly in OMEGAMON II

Using the object-action technique and conventional keys to navigate, you can easily move anywhere in OMEGAMON II. However, as you become more comfortable with the product, you may want to use several powerful navigation techniques.

- mnemonics
- action codes
- fast paths
- pushbuttons

In this section we cover the following topics:

- “Using Mnemonics” on page 33
- “Using Action Codes” on page 34
- “Using Multiple Mnemonics to Create a Fast Path” on page 35
- “Using Pushbuttons” on page 36

Using Mnemonics

We have designated certain letters as mnemonics (one-letter abbreviations) for selections you make. Mnemonics are identified with highlighting, underlining, or parentheses. Once you learn the mnemonics, you can use them to make selections quickly. You will see later how several mnemonics strung together can create an even quicker way of navigating, called fast paths.

Examples of mnemonics

The following table lists several mnemonics from a variety of panels.

<table>
<thead>
<tr>
<th>Action</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Data Collection</td>
<td>l</td>
<td>Controls the frequency at which data is collected for the system Status panel.</td>
</tr>
<tr>
<td>ToolKit</td>
<td>k</td>
<td>Provides Operator’s and System Programmer’s tools to make changes</td>
</tr>
<tr>
<td>Index</td>
<td>i</td>
<td>Accesses panels in the product from a hierarchical menu structure arranged by topic.</td>
</tr>
<tr>
<td>Exit</td>
<td>x</td>
<td>Exits the product.</td>
</tr>
</tbody>
</table>
Using a mnemonic
To use a mnemonic, type the mnemonic in the input field and press Enter.

Using Action Codes
Action codes are mnemonics with special characteristics. Unlike other mnemonics, action codes have the same meanings across panels, and they are operable on most (although not all) panels. They are an alternative for the object-action model, that is, using a slash (/) before an object, then selecting an action from the Actions pull-down menu.

Most commonly used action codes
The following action codes are the most commonly used and are shown on the System Status panel.

<table>
<thead>
<tr>
<th>Action</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Details</td>
<td>s</td>
<td>Shows more detail about the selected item.</td>
</tr>
<tr>
<td>Threshold</td>
<td>l</td>
<td>Allows a setting for each status bar on the System Status panel. This setting is the number (1 - 99 or OFF) of intervals that a threshold must be exceeded before it affects the color of the status bar. This can eliminate short-term problems that cause status bars to change color between refresh intervals.</td>
</tr>
<tr>
<td>Analyze Problems</td>
<td>a</td>
<td>Provides information about: 1) causes of warning or critical conditions; 2) existing threshold settings; 3) existing thresholds compared against current values. <strong>Note:</strong> This action code is available on the System Status panel only.</td>
</tr>
<tr>
<td>Bottlenecks</td>
<td>b</td>
<td>Shows bottlenecks affecting the selected item.</td>
</tr>
<tr>
<td>Historical trends</td>
<td>t</td>
<td>Shows historical trends over a series of time intervals.</td>
</tr>
<tr>
<td>Historical Details</td>
<td>h</td>
<td>Shows historical details over a specified time period.</td>
</tr>
</tbody>
</table>

**Note:** When you use the a action code, OMEGAMON II suspends its auto refresh of the data that caused a warning or critical condition. The data is “frozen” so you can see the exact set of problems that caused the condition at that specific time, even if the problem no longer exists.

Using an action code
To use an action code, type the action code in the input field and press Enter.
Using Multiple Mnemonics to Create a Fast Path

When you become familiar with the product, you may want to take advantage of the time saving feature known as fast paths. A fast path can let you skip intervening panels to reach the desired level of detail quickly by stringing mnemonics together.

Using a fast path

In the input field of the action bar, you can enter a string of up to four mnemonics, taking you down as many as four levels. Mnemonics are identified on menus by highlighting, underlining, or parentheses. Refer to the *OMEGAMON II for MVS Quick Reference* for a list of all the fast paths.

Examples of common fast paths

The following fast paths are frequently used and may prove helpful to you.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Actions Included</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>ot</td>
<td>Options</td>
<td>Thresholds pop-up window(s) appears from which you select the type of threshold that you would like to view. Then a pop-up window showing the thresholds is displayed.</td>
</tr>
<tr>
<td></td>
<td>Thresholds</td>
<td></td>
</tr>
<tr>
<td>go</td>
<td>GoTo</td>
<td>Accesses OMEGAMON for MVS where you can enter OMEGAMON commands.</td>
</tr>
<tr>
<td></td>
<td>OMEGAMON</td>
<td></td>
</tr>
<tr>
<td>ge</td>
<td>GoTo</td>
<td>Accesses EPILOG where you can enter EPILOG reporter commands.</td>
</tr>
<tr>
<td></td>
<td>EPILOG</td>
<td></td>
</tr>
<tr>
<td>ikoe</td>
<td>Index</td>
<td>Displays the Enqueue and Reserve Conflicts panel.</td>
</tr>
<tr>
<td></td>
<td>Toolkit (K)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Operator’s Toolkit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enqueue and Reserve Conflicts</td>
<td></td>
</tr>
<tr>
<td>ocs</td>
<td>Options</td>
<td>Displays the Session Defaults pop-up window.</td>
</tr>
<tr>
<td></td>
<td>Controls</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sessions Defaults</td>
<td></td>
</tr>
<tr>
<td>icse</td>
<td>Index</td>
<td>Displays the System Environment panel, which includes information about your current system configuration, such as RMF and MVS levels, ESCON™ status, high-level LPAR statistics, and the current WLM mode and service policy.</td>
</tr>
<tr>
<td></td>
<td>CPU Information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System Information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System Environment Information</td>
<td></td>
</tr>
</tbody>
</table>
Enabling fast path mnemonics

If you do not see an input field to the left of Actions on the action bar, you will be unable to use fast paths. To display the action bar input field used to accept fast path mnemonics, perform the following steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select <strong>Options</strong> from the action bar.</td>
</tr>
<tr>
<td>2</td>
<td>Enter 2 to select Preferences and display the Preferences pop-up window.</td>
</tr>
</tbody>
</table>
| 3    | Type **On** in the Mmemonics field and press Enter.  
The System Status panel appears, with an action bar input field in the upper left corner of your display screen. |

Using Pushbuttons

OMEGAMON II provides pushbuttons as a way to navigate to panels that are closely related in subject and level of detail to the current panel. They are grouped in a way that offers suggestions regarding where you might turn next for information related to the current panel.

Pushbuttons appear in brackets < > or parentheses () at the bottom of some panels directly above the function key descriptions. Only those in brackets < > are selectable.

**Using a pushbutton**

To use a pushbutton, place the cursor on the phrase or word in brackets < >, and press Enter.

**Note:** The pushbutton for the current panel appears in parentheses and is not selectable.
Pushbutton example

The following panel shows pushbuttons above the function keys.

---

F1=Help  F2=Keys  F3=Exit  F5=Refresh  F6=Console  F10=Action Bar
F11=Print  F15=System Status
Obtaining Online Help

OMEGAMON II has a complete online help system. You can easily access the help system from any input field or any panel, as well as from the Help pull-down menu.

Kinds of help

The following table contains a description of the most important kinds of online help available.

<table>
<thead>
<tr>
<th>Kinds of Help</th>
<th>What to Expect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field level help</td>
<td>A short description of the selected field, and guidelines for what to enter in the field.</td>
</tr>
<tr>
<td>Extended help</td>
<td>A description of an entire panel; how to use the panel, and how you may proceed.</td>
</tr>
<tr>
<td>Glossary</td>
<td>A definition of a technical term.</td>
</tr>
</tbody>
</table>

In this section we cover the following topics.

- “Obtaining Help about Input Fields” on page 38
- “Obtaining Help about Panels” on page 39
- “Looking up a Technical Term” on page 40
- “Looking up a Technical Term” on page 40

Obtaining Help about Input Fields

Field level help is available from any input field.

Example of field level help

If you are on the System Status panel and you want to know what the Batch status light indicates, move the cursor to the input field immediately preceding the Batch status light and press F1. A pop-up help window appears that includes a short description of the Batch status light and what you can do from that input field, as shown below.
Obtaining Help about Panels

Extended help (panel help) is available from any field level help pop-up window, as well as from the Help pull-down menu.

Example of extended help

If you are in the Batch help pop-up window and you want to know about the System Status panel, press F2 and another pop-up window appears that contains information about the panel. This help window gives a description of the panel, explaining both how to use the panel and how to proceed.

The extended help for the System Status panel is shown below.
More+ and More- indicate more help

In the upper right-hand corner of a help pop-up window, you may see the word **More+**, **More-**, or **More+-**. All three of these indicate that you can scroll to more help windows on this topic:

- **More+** indicates that scrolling forward with F8 will produce another help window.
- **More-** indicates that scrolling backward with F7 will produce a previous help window.
- **More+-** indicates that you may scroll forward or backward in the help window.

Looking up a Technical Term

The glossary, which contains definitions of technical terms found in OMEGAMON II, is available from any help pop-up window, as well as any panel (using the Help pull-down menu).
Obtaining the glossary
There are three ways to look up a technical term.

<table>
<thead>
<tr>
<th>IF your location is...</th>
<th>THEN...</th>
</tr>
</thead>
<tbody>
<tr>
<td>the action bar of any panel</td>
<td>enter fast path <strong>hg</strong> to bypass the Help pull-down menu and go straight to the glossary.</td>
</tr>
<tr>
<td>a help pop-up window</td>
<td>press F5.</td>
</tr>
<tr>
<td>a help pop-up window containing a highlighted phrase</td>
<td>tab to the phrase and press Enter for a definition of that phrase.</td>
</tr>
</tbody>
</table>

Example of the glossary
Suppose you are reading through the Batch help pop-up when you see the phrase “elapsed time”, a phrase you are unfamiliar with. You press F5 to access the glossary.
The glossary pop-up window appears. Press F6 to start your search. When you press F6, the Search Glossary pop-up window appears.

Type “elapsed” in the Search Term line and press Enter. The search term “elapsed” opens the glossary to reveal two dictionary items, as shown.
Select “elapsed” from this pop-up window to access the glossary definition. The glossary definition of “elapsed time” appears.
Other Helpful Information

Other helpful information can be obtained using the help function keys and the Help pull-down menu.
Help function keys

The following describes the help function keys and their assignments. Some of the assignments are unique to the OMEGAMON II help system.

<table>
<thead>
<tr>
<th>Function key</th>
<th>Name</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Field level help</td>
<td>Displays field level help if pressed on an entry field, extended help if pressed anywhere else.</td>
</tr>
<tr>
<td>F2</td>
<td>Extended help</td>
<td>Displays general information about the current panel or pop-up.</td>
</tr>
<tr>
<td>F3</td>
<td>Exit</td>
<td>Exits the help system.</td>
</tr>
<tr>
<td>F5</td>
<td>Glossary</td>
<td>Displays a list of glossary terms and phrases.</td>
</tr>
<tr>
<td>F6</td>
<td>Search</td>
<td>Permits a search of the glossary or help index for all terms and phrases containing the word you’ve specified.</td>
</tr>
<tr>
<td>F7</td>
<td>Backward</td>
<td>Scrolls backward.</td>
</tr>
<tr>
<td>F8</td>
<td>Forward</td>
<td>Scrolls forward.</td>
</tr>
<tr>
<td>F9</td>
<td>Function keys help</td>
<td>Describes the assignments for each function key.</td>
</tr>
<tr>
<td>F11</td>
<td>Index</td>
<td>Displays the help index.</td>
</tr>
<tr>
<td>F12</td>
<td>Cancel</td>
<td>Cancels the current window.</td>
</tr>
</tbody>
</table>

Using the help pull-down menu

The following is an example of the Help pull-down menu, showing all the kinds of help available. You can access the Help pull-down menu by entering an h in the action bar input field.
Other help features

The Help pull-down menu has other features that were not described earlier in this section: Help for Help, Tutorial, About, User Information, and New Features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help for Help</td>
<td>Tells you about the OMEGAMON II help features and how to access them.</td>
</tr>
<tr>
<td>Tutorial</td>
<td>Teaches you how to move around OMEGAMON II, how to use parts of the screen, and how to access the help system.</td>
</tr>
<tr>
<td>About</td>
<td>Gives you the Candle copyright screen.</td>
</tr>
<tr>
<td>User Information</td>
<td>Gives you information about the user currently logged onto this terminal: user ID, terminal ID, and system ID.</td>
</tr>
<tr>
<td>New Features</td>
<td>Tells you the features that have been added to OMEGAMON II since the last release.</td>
</tr>
</tbody>
</table>
Introduction

Once you have been running OMEGAMON II for a while and are familiar with the kinds of information presented on its panels, you may want to change the content of the panels so they reflect information that is more appropriate to the task you are performing.

This chapter describes how to adjust your OMEGAMON II environment by changing the OMEGAMON II defaults your OMEGAMON II Customizer has set up for you.

Chapter Contents

- Controlling the Appearance and Content of Panels ......................... 48
- Adjusting Thresholds ........................................................................ 53
- Understanding OMEGAMON II Profiles ........................................... 56
- Setting OMEGAMON Logging Activity On or Off ............................ 60
- Synchronizing OMEGAMON II to Local System Time ..................... 61
Controlling the Appearance and Content of Panels

Introduction

You can control the appearance and content of OMEGAMON II panels by:

- controlling the format of certain display fields
- specifying navigation options
- indicating the kinds of information you want to see displayed on a panel
- controlling the frequency with which information is collected for a panel

Although you can control the format and content of display fields on a panel, you cannot customize column headers themselves.

This section, which describes the procedures necessary to achieve these results, covers the following topics.

- Controlling the Appearance of Panels
- Controlling the Content of Panels

Controlling the Appearance of Panels

You can customize the appearance of information on OMEGAMON II panels by:

- specifying the format of certain display fields, such as date and time suppressing or activating certain information, such as panel ID and function key area
- specifying navigation options between OMEGAMON II, OMEGAMON, and EPILOG, and between OMEGAMON II panels themselves
Enabling fast path mnemonics
Throughout this section, and in other parts of this manual, fast paths are provided to enable you to quickly access the panels and pop-ups used to perform certain tasks. If you do not see an input field to the left of Actions on the action bar, you will not be able to enter these fast paths. To display the action bar input field upon which you enter fast path mnemonics, perform the following steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select Options from the action bar.</td>
</tr>
<tr>
<td>2</td>
<td>Enter 2 to select Preferences and display the Preferences pop-up window.</td>
</tr>
<tr>
<td>3</td>
<td>Type On in the Mnemonics field and press Enter. The System Status panel appears, with an action bar input field in the upper left corner of your display screen.</td>
</tr>
</tbody>
</table>

You are now ready to navigate through OMEGAMON II using fast paths.

Controlling formats
The following table describes the information formats you can control on an OMEGAMON II panel.

| If you want to...                  | THEN enter fast path... | AND fill in this field (press F1 for help)...
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>display characters with or without color on the System Status panel</td>
<td>ocs</td>
<td>Use status characters?</td>
</tr>
<tr>
<td>identify the character to use for critical conditions</td>
<td>ocs</td>
<td>Critical status character</td>
</tr>
<tr>
<td>identify the character to use for warning conditions</td>
<td>ocs</td>
<td>Warning status character</td>
</tr>
<tr>
<td>identify the character to use for normal conditions</td>
<td>ocs</td>
<td>Okay status character</td>
</tr>
<tr>
<td>specify the format in which dates appear</td>
<td>op</td>
<td>Date format</td>
</tr>
<tr>
<td>specify the format in which times appear</td>
<td>op</td>
<td>Time format</td>
</tr>
<tr>
<td>specify the character to appear between hours, minutes, and seconds</td>
<td>op</td>
<td>Time separator character</td>
</tr>
<tr>
<td>specify the character to appear between hours, minutes, and seconds</td>
<td>op</td>
<td>Time morning character</td>
</tr>
<tr>
<td>specify the characters to indicate afternoon hours on a 12-hour clock</td>
<td>op</td>
<td>Time afternoon indicator</td>
</tr>
</tbody>
</table>
Suppressing or activating information
The following table describes the kinds of information you can suppress or activate on an OMEGAMON II panel.

<table>
<thead>
<tr>
<th>IF you want to suppress or activate the...</th>
<th>THEN enter fast path...</th>
<th>AND enter On or Off in this field...</th>
</tr>
</thead>
<tbody>
<tr>
<td>terminal alarm for invalid input</td>
<td>op</td>
<td>Beep</td>
</tr>
<tr>
<td>panel ID for current panel</td>
<td>op</td>
<td>Panel ID</td>
</tr>
<tr>
<td>current time and date in the upper right corner of every panel</td>
<td>op</td>
<td>Time/date display</td>
</tr>
<tr>
<td>list of currently available function keys</td>
<td>op</td>
<td>Function key area</td>
</tr>
</tbody>
</table>

Controlling navigation
The following table describes the kinds of navigation you can control between OMEGAMON II, panels and between OMEGAMON II and OMEGAMON and EPILOG.

<table>
<thead>
<tr>
<th>IF you want to...</th>
<th>THEN enter fast path...</th>
<th>AND fill in this field (press F1 for help)...</th>
</tr>
</thead>
<tbody>
<tr>
<td>use the Enter key as an alternative to the s action code</td>
<td>op</td>
<td>Implicit action</td>
</tr>
<tr>
<td>set the key used to return to OMEGAMON II from OMEGAMON or EPILOG</td>
<td>ocs</td>
<td>Trigger to return from zoom</td>
</tr>
<tr>
<td>supply the realtime password so you can access password-protected OMEGAMON II panels and OMEGAMON commands</td>
<td>oct</td>
<td>Password (and user ID, if applicable for external security)</td>
</tr>
</tbody>
</table>

Controlling the Content of Panels
You can control the content of OMEGAMON II panels by:

- specifying the kinds of information displayed on a panel, such as bottleneck analysis information
- controlling the frequency with which information is collected for a panel
Controlling kinds of information

The following table describes the panel content you can control.

<table>
<thead>
<tr>
<th>IF you want to...</th>
<th>THEN enter fast path...</th>
<th>AND fill in this field (press F1 for help)...</th>
</tr>
</thead>
<tbody>
<tr>
<td>view all information or only problems</td>
<td>ocs</td>
<td>Default view</td>
</tr>
<tr>
<td>specify the OMEGAMON environment to use when entering command language commands</td>
<td>ocs</td>
<td>Zoomed-to OMEGAMON user profile</td>
</tr>
<tr>
<td>select wait reasons on which to report</td>
<td>ocw</td>
<td>select individual wait reasons</td>
</tr>
<tr>
<td>control collection of bottleneck analysis information</td>
<td>ocs</td>
<td>bottleneck analysis</td>
</tr>
<tr>
<td>control collection of bottleneck analysis information</td>
<td>ocs</td>
<td>Enable impact analysis</td>
</tr>
</tbody>
</table>

Controlling frequency of collection

The following table describes the kinds of collection frequencies you can control.

<table>
<thead>
<tr>
<th>IF you want to...</th>
<th>THEN enter fast path...</th>
<th>AND fill in this field (press F1 for help)...</th>
</tr>
</thead>
<tbody>
<tr>
<td>continue impact analysis data collection for a certain period of time</td>
<td>ocs</td>
<td>Impact analysis time-out period</td>
</tr>
<tr>
<td>control the frequency with which new information is displayed on a panel automatically</td>
<td>ocs</td>
<td>Auto refresh interval (specify a number)</td>
</tr>
<tr>
<td>suspend auto refresh</td>
<td>ocs</td>
<td>Auto refresh interval (set to Off)</td>
</tr>
</tbody>
</table>

Note: This prevents OMEGAMON II from automatically displaying information; you can press F5 at any time to manually refresh the information on a panel.

Controlling data collection frequency for an object

In addition to controlling the frequency with which new information is automatically displayed on a panel (the auto refresh interval), you can control the frequency with which data is collected for individual objects on the System Status panel body.

If an object is important to you at this time, you can request that data be collected for it with every auto refresh. If, however, some object is important now, you may choose to collect data for this object less frequently for the time being, thus reducing the impact on system resources.
Changing the data collection frequency for an object

To change the data collection frequency, enter l (lower case L) in the input field to the left of any object on the panel body of the System Status panel. OMEGAMON II displays a pop-up window that allows you to control the data collection frequency for the selected object.
Adjusting Thresholds

In addition to controlling the appearance and content of panels, you can adjust the performance thresholds that have been set for your system.

What is a threshold?
A *threshold* is a value or condition against which OMEGAMON II measures your system’s performance. When the performance of a workload or resource exceeds a threshold, OMEGAMON II displays a warning (yellow) or critical (red) status indicator on the System Status panel.

In this section we cover the following topics.
- "Analyzing Existing Thresholds," below
- “Adjusting Thresholds” on page 54

Analyzing Existing Thresholds
The OMEGAMON II Customizer defines a common set of thresholds for all users at your site. Some groups or individual users, such as systems programmers who may be charged with anticipating particular kinds of complaints before other OMEGAMON II users, may have different requirements for threshold settings to perform their jobs. Users can override the default settings for their own environment using the Options pull-down.

When should you change threshold settings?
You might want to change threshold settings when:
- a workload or resource repeatedly reflects a warning or critical condition when the situation is not significant to you
- a workload or resource displays a warning condition too late for you to take action to prevent a critical condition
- a service-level agreement changes
- your role requires a different set of thresholds from the defaults your OMEGAMON II Customizer has set up for you
Determining whether your settings are appropriate

You can use the Analyze Problems panel (accessed by entering A next to a yellow or red status indicator on the System Status panel), in combination with information from historical panels, to help you determine whether your current threshold settings are appropriate. In addition to your current threshold settings, the Analyze Problems panel displays all activity contributing to a warning or critical condition for a workload, resource, or operator alert. You can print this panel, as well as historical trend and detail panels, to keep a record of recurring warning or critical problems and use this record to decide upon a new threshold when necessary.

Setting printing options

This procedure applies to printing OMEGAMON II panels on your mainframe printer. Perform these steps once; your print routing options are saved from one OMEGAMON II session to the next.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter fast path oco to access the Screen Print Output Routing Options panel.</td>
</tr>
<tr>
<td>2</td>
<td>Supply the print options and routing information.</td>
</tr>
<tr>
<td>3</td>
<td>Press Enter to make your changes permanent, then press F12 to exit out of the Options path.</td>
</tr>
</tbody>
</table>

Printing OMEGAMON II panels

The following procedure enables you to print OMEGAMON II panels on your mainframe printer:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Press F11 on the panel you want to print. OMEGAMON II sends the panel image to a print log file. If the panel is longer than one screen of data, a pop-up panel appears to ask if you want to print the entire panel. Select option 1 from the pop-up menu, and press Enter to print the panel image as you see it. Select option 2 from the pop-up menu, and press Enter to print the entire panel image.</td>
</tr>
<tr>
<td>2</td>
<td>Enter fast path ol to close the print log file and print the panel images contained in it.</td>
</tr>
</tbody>
</table>

Adjusting Thresholds

After you have determined that some of your existing thresholds need to be changed, you can change them through the Thresholds menu.

Thresholds: fast path

To access the Thresholds menu, enter fast path ot.
Migrating thresholds from OMEGAMON
If you have already been running OMEGAMON for MVS on your system and you want to use these exceptions (thresholds) in OMEGAMON II, enter fast path `otf`. If you want to make additional changes to these thresholds, log off OMEGAMON II, log on again, and use the fast paths described below.

Changing thresholds
The following table describes where to change each type of OMEGAMON II threshold.

<table>
<thead>
<tr>
<th>To change thresholds for...</th>
<th>ENTER fast path</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource utilization and operator alerts</td>
<td><code>otu</code></td>
</tr>
<tr>
<td>DASD groups</td>
<td><code>otd</code></td>
</tr>
<tr>
<td>performance groups</td>
<td><code>otp</code></td>
</tr>
<tr>
<td>service class (MVS/SP 5 goal-mode users only)</td>
<td><code>ots, oti</code></td>
</tr>
<tr>
<td>domains</td>
<td><code>otm</code></td>
</tr>
<tr>
<td>response time groups</td>
<td><code>otr</code></td>
</tr>
<tr>
<td>critical jobs or tasks</td>
<td><code>otc</code></td>
</tr>
<tr>
<td>critical DASD devices</td>
<td><code>otv</code></td>
</tr>
<tr>
<td>enqueue exclusion list</td>
<td><code>ote</code></td>
</tr>
</tbody>
</table>

Guidelines for changing thresholds
The following are guidelines for changing thresholds.

<table>
<thead>
<tr>
<th>IF you want to...</th>
<th>THEN...</th>
</tr>
</thead>
<tbody>
<tr>
<td>disable a threshold</td>
<td>type Off</td>
</tr>
<tr>
<td>reset a threshold to its default (last saved) setting</td>
<td>clear the field</td>
</tr>
<tr>
<td>display help about a threshold</td>
<td>press F1</td>
</tr>
<tr>
<td>save your changes for a particular threshold group</td>
<td>press Enter, then press F12 to exit.</td>
</tr>
</tbody>
</table>

Note: When you press Enter, the settings on the current pop-up are saved in your user profile, whether you changed them or not.
Understanding OMEGAMON II Profiles

Introduction
Your OMEGAMON II Customizer has set up a default OMEGAMON II environment for you. The settings that determine what this environment looks like are saved in a default profile. All users start out with the same default profile when they first sign on to OMEGAMON II.

When you use the procedures described in this chapter to change the default environment, OMEGAMON II creates a user profile for you and saves the settings you have changed in it. These settings that you changed override the default settings; however, any settings you have not changed are still picked up from the default profile.

Changing your view options on a panel causes a pop-up panel to appear that asks if you want to save your new options in your profile when you log off. If you reply yes on the pop-up panel, your changes are saved to your user profile. The changes will be available in that profile the next time you log on. If you reply no, the changes to your view options remain in effect until you log off.

Maintaining profiles: fast path
You switch, copy, print, and delete profiles through the Profile Maintenance menu. The fast path for this menu is ocp.

Restrictions
Your ability to perform Profile Maintenance functions depends on the access your OMEGAMON II Customizer has given you.

Using different profiles
If you are authorized to create and copy profiles, you can maintain more than one user profile and switch to the different profiles at any time during your OMEGAMON II session. Here are some examples in which this might be useful:

- If your job entails shift work, and your hours cross over from one shift to the next, the settings that are appropriate for one shift may not be appropriate for the next. You can maintain two profiles containing settings appropriate for each shift, and manually switch to the second profile when the next shift begins.

- As a systems programmer, you might receive a call from a user claiming there is a problem with the CPU. Your System Status panel indicates a green CPU status light, however. You can switch to the user’s profile to help see the problem from his or her point of view.

- A help desk employee may want different views of the system at different times. For instance, knowledge of TSO address space activity may be more important at the moment than batch job activity; therefore, a profile that concentrates on TSO activity and deactivates settings for batch jobs might be appropriate.
Copying settings from another profile

You may know of another user whose user profile settings are close to the settings you need to perform your job. Assuming your OMEGAMON II Customizer has given you write access to profiles, you can copy that user’s profile and then modify the copy at will. Sometimes, you may want to copy that other profile on top of an existing profile, overwriting the contents of the existing profile. At other times, you may want to give the copy a new name, thus creating a new profile.

To copy the settings from another profile, perform the following steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
</table>
| 1 | Make your choice:  
  - If you want to copy settings on top of your current profile, proceed to Step 2.  
  - If not, enter fast path `ocps` to identify the profile you want to receive the copied settings. You may enter the name of an existing profile or a new one, if you want to create a new profile. | Your current profile will receive the copied settings in Step 3.  
The specified profile becomes your current profile and will receive the copied settings in Step 3. |
| 2 | Enter fast path `ocpc`. | The Copy a Profile pop-up appears |
| 3 | Supply the name of the profile whose settings you want to copy and press Enter. | A message displays, informing you that the profile settings have been copied to your current user profile. These settings will take effect as soon as you exit out of the Options path. |

Methods for using different profiles

The following table describes two ways to specify different profiles. The profiles must already exist, and may have been created by you or by another user.

<table>
<thead>
<tr>
<th>IF you want to...</th>
<th>THEN enter fast path...</th>
<th>AND supply...</th>
</tr>
</thead>
</table>
| switch to another profile temporarily  
(until you switch it again or end your OMEGAMON II session) | `ocps` | the name of the preferred profile. |
| specify a different startup profile for future OMEGAMON II sessions  
**NOTE:** A user’s startup profile is the profile that is automatically in effect when a user logs onto OMEGAMON II. | `ocph` | the name of the profile to use as your new startup profile. |
Printing a profile
You may want to print a profile to see all of its threshold settings in one place. To print a profile, perform the following steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter fast path <code>ocpp</code>.</td>
<td>The Print a Profile pop-up appears.</td>
</tr>
<tr>
<td>2</td>
<td>Supply the name of the profile you would like to print and press Enter.</td>
<td>The Profile Print Output Routing Options pop-up appears.</td>
</tr>
<tr>
<td>3</td>
<td>Type your printer destination information and press Enter.</td>
<td>A message appears, indicating that the specified profile has been printed.</td>
</tr>
</tbody>
</table>

Deleting a profile
To delete a profile that is no longer needed, perform the following steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter fast path <code>ocpd</code>.</td>
<td>The Delete a Profile pop-up appears.</td>
</tr>
<tr>
<td>2</td>
<td>Supply the name of the profile you would like to delete and press Enter.</td>
<td>A message appears, indicating that the specified profile has been deleted.</td>
</tr>
</tbody>
</table>

**Note:** Only those users who have been granted access privileges by their OMEGAMON II Customizer are able to delete a profile.

Using the Dynamic Profile Update Facility
The Dynamic Profile Update Facility lets you dynamically import and export individual OMEGAMON II profiles to and from a partitioned dataset by creating import and export dialogs that are invoked externally through MVS.

Following are the instructions for using the import and export dialogs:

- Invoke KM2EXP and KM2IMP by using the MVS Modify command.
- Use KM2EXP to export a profile to a partitioned dataset and KM2IMP to import a profile from a partitioned dataset.
- When you export a profile, its data is written in a standard format to a partitioned dataset member.
- The partitioned dataset should be allocated as fixed blocked, LRECL of 80, and BLKSIZE of 8880, with an appropriate number of directory blocks.
- Make sure CANSM2, the OMEGAMON II for MVS started task name, has write authority to the partitioned dataset.
- Importing reverses the above process, and recreates a profile based on the exported information.
The MVS Modify command for exporting a profile is as follows:

```
/F CANSM2,NTD KM2EXP 'PDSNAME MEMBER PROFILE'
```

Where `PDSNAME` is the name of the target partitioned dataset name, `MEMBER` is the target partitioned dataset member, and `PROFILE` is the source profile.

The MVS Modify command for importing a profile is as follows:

```
/F CANSM2,NTD KM2IMP 'PDSNAME MEMBER PROFILE'
```

Where `PDSNAME` is the name of the source partitioned dataset name, `MEMBER` is the source partitioned dataset member, and `PROFILE` is the target profile.
Setting OMEGAMON Logging Activity On or Off

Introduction

OMEGAMON II allows you to control the logging activity between OMEGAMON and OMEGAMON II. When the log is activated, all traffic between OMEGAMON and OMEGAMON II will be logged in a dynamically allocated SYSOUT file in the OMEGAMON PROC.

This section describes how to activate or deactivate logging between OMEGAMON and OMEGAMON II.

Controlling the OMEGAMON log

Logging activity between OMEGAMON and OMEGAMON II is controlled using a fast path command. Enter IKS at the action bar input field, then enter T at the next action bar input field to reverse the traffic logging in effect. If traffic logging is OFF, enter T at the action bar input field to turn it ON. If traffic logging is ON, enter T at the action bar input field to turn it OFF.


Synchronizing OMEGAMON II to Local System Time

Introduction

If the local system time has been reset and you keep OMEGAMON II running, you need to synchronize the time for OMEGAMON II to the new time. Local time is normally reset when correcting the TOD IPL setting or when switching between standard time and daylight savings time.

Resetting the time for OMEGAMON II

To synchronize the time for OMEGAMON II with the local system time, enter the following MVS operator commands:

\[ F \text{CANSM2}, \text{TIME RESET:exmp.} \]

where CANSM2 is the OMEGAMON II started task.

\[ F \text{cms}, \text{TIME RESET} \]

where cms is the Candle Management Server (CMS) started task.

Note: To designate an address space in which to run the started task, add a number from 00-15 to the name of the started task. For example, CANSM203 specifies 03 as the address space.

Resetting the time for other Candle applications

All Candle applications using the same CMS as OMEGAMON II need to have their time reset. This is done by entering the following MVS operator command for each application:

\[ F \text{application}, \text{TIME RESET} \]

where application is the appropriate Candle application.

Resetting the time with the historical collector running

If you reset the local system time earlier than the current time and the historical collector is running, you will get VSAM errors. The collector considers it an error to be writing records for a time period that has already been recorded. In this situation you will need to stop the collector. To stop the historical collector, enter the following MVS operator command:

\[ P \text{CANSM2HI} \]

where CANSM2HI is the historical collector started task.

Once enough time has elapsed so that duplicate records will not be created, you can restart the historical collector. To restart the historical collector, enter the following MVS operator command:

\[ S \text{CANSM2H} \]

where CANSM2HI is the historical collector started task.
Synchronizing OMEGAMON II to Local System Time
Section 1
IPS-Based Performance Monitoring
Monitoring System Performance

Introduction

The OMEGAMON II System Status panel is the starting point for monitoring your MVS system. To resolve performance problems on your system you need to understand how to interpret the status indicators, navigate through the product in response to status alerts, and track a problem to its source.

The examples presented in this chapter are designed to help you learn how to monitor your system.

Chapter Contents

Interpreting Status Lights ................................................................. 66
Monitoring Workloads, Resources, and Alerts: ................................... 71
Monitoring Performance across Systems ............................................ 129
Monitoring Historical Shared-DASD Performance .............................. 134
Interpreting Status Lights

Introduction

The System Status panel compares the current performance of monitored objects to their threshold values. The thresholds are based on service objectives at your site, as defined by your site’s OMEGAMON II Customizer. If a problem exists, a light or symbol appears in a status indicator area dynamically alerting you to the problem.

In this section we cover the following topics.

- Interpreting Status Lights
- Monitoring Workloads, Resources, and Alerts
- Monitoring Performance across Systems

The panel body of the System Status panel is organized into three status indicator areas:

- Workload status
- Resource status
- Operator Alerts

Within the status indicator areas, colors or symbols are used to signal status alerts. In the screen below, we use symbols ($$$$$, *****, and -----) to show the status of the monitored objects in the panel body.
Colored lights and associated conditions

On a color terminal, each item on the panel body can be any one of four colors, and each color signals a different condition of the system.

The four colors and the conditions they indicate are shown below.

<table>
<thead>
<tr>
<th>Color</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>Normal condition</td>
</tr>
<tr>
<td>Yellow</td>
<td>Warning condition (there is a possible system problem)</td>
</tr>
<tr>
<td>Red</td>
<td>Critical condition (there is a definite system problem)</td>
</tr>
<tr>
<td>Turquoise</td>
<td>Monitoring of object is idle (data is not available because RMF is inactive) or disabled (using the L action code).</td>
</tr>
</tbody>
</table>

Symbols and associated conditions

If your terminal does not support extended attributes or is not a color terminal, symbols instead of colors are used as status indicators.

The symbols and conditions indicated are shown below.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyphen (-)</td>
<td>Normal condition</td>
</tr>
<tr>
<td>Asterisk (*)</td>
<td>Warning condition</td>
</tr>
<tr>
<td>Dollar sign ($)</td>
<td>Critical condition</td>
</tr>
<tr>
<td>Blanks</td>
<td>Monitoring of object is idle or disabled</td>
</tr>
</tbody>
</table>

Note: The symbols above are the default settings. You can change the default settings of the symbols if you wish.

- To change the default settings, select the Options pull-down menu. Select Controls from the Options pull-down. Then select Session defaults and change the appropriate fields.

General Guidelines for Handling Status Alerts

On the following pages we present two tables for handling status alerts:

- for workloads or resources (left side of the System Status panel)
- for operators (right side of the System Status panel)

In both tables we present actions you might take to meet a variety of objectives. As a reminder for new users, start with s for Show Details as a first step and then proceed through the other steps.
As you become more familiar with the product, you will learn where Bottlenecks or Analyze Problems are productive alternatives as a first step.

**Responding to status alerts for workloads or resources**

You can investigate the status of any object on the panel at any time no matter what color the status light is.

But, when you see a red or yellow status light for workload or resource status on the left side of the System Status panel, the following suggestions for action may be useful to you in your investigation.

<table>
<thead>
<tr>
<th>IF the objective is to see...</th>
<th>THEN...</th>
<th>TO see this result...</th>
</tr>
</thead>
<tbody>
<tr>
<td>details about an object</td>
<td>type s for Show Details on the input line before the item</td>
<td>a panel of details that allows you to begin to isolate the problem area.</td>
</tr>
<tr>
<td>a list of conditions that exceed thresholds</td>
<td>type a for Analyze Problems</td>
<td>a comparison between actual and expected performance for a specific indicator.</td>
</tr>
<tr>
<td>the trends in performance for an object in the past</td>
<td>type h for Historical Trends for the past performance</td>
<td>how a workload or resource has performed over a number of intervals.</td>
</tr>
<tr>
<td>how an object has performed within a specified time period</td>
<td>type h for Historical Details for past performance</td>
<td>detailed information for a selected time period combined into a single averaged result.</td>
</tr>
<tr>
<td>whether a bottleneck exists</td>
<td>type s for Show Details to take you one level below the System Status panel. Then type b for Bottlenecks on the input line before the job or performance group</td>
<td>the Bottlenecks panel, enabling you to identify conditions affecting workload performance.</td>
</tr>
</tbody>
</table>

**Note**: Bottleneck analysis is appropriate for objects such as address spaces, for example.
Interpreting Status Lights

Responding to status alerts for operators

You can investigate the status of any object on the panel at any time no matter what color the status light is.

But, when you see a red or yellow status light for operator alerts on the right side of the System Status panel, the following suggestions for action may be useful to you in your investigation.

<table>
<thead>
<tr>
<th>IF the objective is to see...</th>
<th>THEN...</th>
<th>TO see this result...</th>
</tr>
</thead>
<tbody>
<tr>
<td>details for an object</td>
<td>type s for Show details on the input line before the object</td>
<td>for all indicators except HSM, SMF, and Enqueue, a message indicating the problem. For HSM, SMF, and Enqueue, a detail panel is displayed</td>
</tr>
<tr>
<td>the System Console</td>
<td>press Enter on the object input line and then press Enter again from the pop-up window (or press F6)</td>
<td>the System Console, enabling you to enter MVS or other commands.</td>
</tr>
<tr>
<td>additional related information</td>
<td>type g for GoTo on the input line of the action bar</td>
<td>the GoTo pull-down menu, enabling you to choose related information that might help you diagnose the problem.</td>
</tr>
<tr>
<td>closely related information about the same object</td>
<td>place the cursor on the appropriate pushbutton available at the bottom of some panels (not the System Status panel) and press Enter</td>
<td>panels closely related in subject and level of detail to the current panel.</td>
</tr>
</tbody>
</table>
General Guidelines for Monitoring a Healthy System

There are times when you want to investigate some aspect of a particular object’s performance, though at the time no red or yellow status light is showing for the object.

This section gives some guidelines for investigating objects that show only green (for normal) or turquoise (for idle) on the System Status panel.

The same navigating concepts apply regardless of the color of the status light. You navigate around the system the same way whether the status lights are green for normal, yellow for warning, or red for critical.

Investigating a healthy system

The table that follows presents some general guidelines for exploring an object with OMEGAMON II though its status light is green.

<table>
<thead>
<tr>
<th>IF you want to...</th>
<th>THEN:</th>
</tr>
</thead>
<tbody>
<tr>
<td>see a panel of details about an object</td>
<td>type s for Show Details on the input field before the object.</td>
</tr>
<tr>
<td>see the trends in performance for an object over a period of time</td>
<td>type t for Historical trends on the input field before the object.</td>
</tr>
<tr>
<td>see detailed information on how an object has performed over a selected time period, combined into a single averaged result</td>
<td>type h for Historical Details on the input field before the object.</td>
</tr>
<tr>
<td>see additional related information on the GoTo pull-down menu</td>
<td>type g for GoTo on the input line of the action bar.</td>
</tr>
<tr>
<td>see closely-related information about an object</td>
<td>place the cursor on the appropriate pushbutton at the bottom of a panel, where available, and press Enter. Accessible pushbutton panels are enclosed in &lt;   &gt;.</td>
</tr>
</tbody>
</table>
Monitoring Workloads, Resources, and Alerts:

Introduction

In this section we present examples of how you might use OMEGAMON II to monitor your system’s performance. We have included at least two examples from each of the three status indicator areas.

- Workload status - Six examples: current and recurring TSO response time problems, excessive elapsed time, I/O wait, and batch and STC problems
- Resource status - Five examples: CSA problem, cache statistics (DASD-related), replacing link list datasets, and channel I/O and CPU utilization problems
- Operator Alerts - Two examples: key task and enqueue conflict

The procedures used in the examples do not vary greatly from the procedures you would use for any of the other objects found on the System Status panel. We encourage you to go through each example to learn how to monitor performance in each status indicator area.

In this section we cover the following topics:

- “Example: Investigating a Current Response Time Problem” on page 72
- “Example: Investigating a Recurring Response Time Problem” on page 76
- “Example: Investigating Excessive Elapsed Time” on page 80
- “Example: Resolving an I/O Wait Problem” on page 85
- “Example: Resolving a Batch Problem” on page 93
- “Example: Resolving a Started Task Problem” on page 95
- “Example: Investigating a CSA Problem” on page 101
- “Example: Looking Up Cache Statistics” on page 105
- “Example: Replacing Link List Datasets” on page 110
- “Example: Resolving a Channel I/O Problem” on page 115
- “Example: Resolving a CPU Utilization Problem” on page 122
- “Example: Investigating a Key Task Alert” on page 124
- “Example: Responding to an Enqueue Conflict” on page 126
Example: Investigating a Current Response Time Problem

If your thresholds are set correctly for your site, you should see a TSO response time problem on the System Status panel before anyone calls the data center to complain about it.

Example of poor TSO response time

Suppose you are monitoring your system from the System Status panel, and you see the TSO: RTA™ status light turn red, as pictured in the following figure.

To investigate the cause of the red light, you issue the Show Details action for this indicator by moving the cursor to the TSO: RTA input field and pressing Enter.

Issuing the Show details action for the TSO: RTA indicator on the System Status panel leads you to the TSO Response Time Groups panel pictured below.
In this example, the San Francisco group has an unusually high host response time of 25.63 seconds. You enter s in the San Francisco input field to display the TSO Response Time Group Users panel, as shown in the following figure.

You notice that user TSR264A has a high host response time of 48.88 seconds.
You enter s in the TSR264A input field to display the Details for a TSO User panel, as shown in the following figure.

You notice a 50-second time in transaction and high CPU wait of 93% for Period 2.

You enter s in the CPU wait input field to display the Resource Bottleneck panel, as shown in the following figure.

You notice a test job is the main bottleneck since over 72 percent of the time TSR264A was ready to use the CPU and the test job was preventing it from doing so.
You decide to reset the test job to the sleeper performance group. Notify the person who submitted this job and explain that this job has been set to the sleeper performance group and this will lower this job's priority.
Example: Investigating a Recurring Response Time Problem

The following example shows how you may use the historical approach to resolve a recurring TSO response time problem.

Example of recurring TSO response time problem

Suppose you receive a complaint from the users in performance group 2 that response time is poor between 3:30 and 5:00 PM every day. You need to find out why. You request the **Historical trends** action for TSO Host on the System Status panel.

Type the number of the problem TSO performance group (in this example, group number 2) in the pop-up window, and press Enter.
Monitoring Workloads, Resources, and Alerts:

This displays the Trend Date/Time Selection pop-up window, which offers you a selection of date and time ranges for the trend data display.

<table>
<thead>
<tr>
<th>Workload status</th>
<th>Resource status</th>
<th>Operator Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>CPU</td>
<td>Key Task</td>
</tr>
<tr>
<td>STC/APPC</td>
<td>DASD</td>
<td>WTO Buffer</td>
</tr>
<tr>
<td>TSO: RTA</td>
<td>Tape</td>
<td>WTOs</td>
</tr>
<tr>
<td>t TSO Host</td>
<td>Paging</td>
<td>OLTEP</td>
</tr>
<tr>
<td>A</td>
<td>KM2HIRNG</td>
<td>Trend Date/Time Selection</td>
</tr>
<tr>
<td>D</td>
<td>Select a range for the display of trend data.</td>
<td></td>
</tr>
</tbody>
</table>

- 4: Last 4 hours
- 2: Last hour today and same time for yesterday
- 3: Last hour today and every day since Monday
- 4: Advanced date and time options...

Save date/time settings in profile ... No No/Yes
Suppress this popup on Trend panels ... No No/Yes

F1=Help  F4=Prompt  F12=Cancel

For this scenario, you want to display information for the 3:30 to 5:00 PM time period, so you select option 4 and press Enter.
This displays the Trend Date/Time Settings pop-up window. Set your desired date and time settings (3:30 to 5:00 every day for the last few days) on the Trend Date/Time Settings pop-up window.

Enter your time settings in military time, 15:30 to 17:00, and press Enter.
This displays the Historical Trends for a TSO Performance Group panel.

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>N Trans</th>
<th>Main Bottleneck (*)</th>
<th>Avg Resp</th>
</tr>
</thead>
<tbody>
<tr>
<td>_08/22</td>
<td>16:59</td>
<td>106</td>
<td>Private Page-In Wa</td>
<td>.08 s</td>
</tr>
<tr>
<td>_08/22</td>
<td>16:44</td>
<td>105</td>
<td>Private Page-In Wa</td>
<td>.09 s</td>
</tr>
<tr>
<td>_08/22</td>
<td>16:14</td>
<td>117</td>
<td>Private Page-In Wa</td>
<td>.16 s</td>
</tr>
<tr>
<td>_08/22</td>
<td>15:59</td>
<td>101</td>
<td>Private Page-In Wa</td>
<td>.05 s</td>
</tr>
<tr>
<td>_08/22</td>
<td>15:44</td>
<td>140</td>
<td>Private Page-In Wa</td>
<td>.19 s</td>
</tr>
<tr>
<td>_08/21</td>
<td>16:59</td>
<td>350</td>
<td>Private Page-In Wa</td>
<td>.20 s</td>
</tr>
<tr>
<td>_08/21</td>
<td>16:29</td>
<td>107</td>
<td>STIMER Wait</td>
<td>.07 s</td>
</tr>
<tr>
<td>_08/21</td>
<td>16:14</td>
<td>267</td>
<td>Long Wait</td>
<td>.13 s</td>
</tr>
</tbody>
</table>

As the above example shows, the main bottleneck for most intervals is private page-in wait. You decide to investigate further, to identify the causes of the main bottleneck.

**Identifying causes of the main bottleneck**

To find causes for the main bottleneck, show details for real storage use and paging devices, and check your System Resources Manager (SRM) values. These are some of the resources that you would want to look at to determine the cause of the bottleneck.

<table>
<thead>
<tr>
<th>Panel</th>
<th>Fast Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Storage Use by Virtual Areas</td>
<td>ispr</td>
</tr>
<tr>
<td>Paging Activity</td>
<td>isap</td>
</tr>
<tr>
<td>SRM Information</td>
<td>ikss</td>
</tr>
</tbody>
</table>

Once you determine the cause of the bottleneck, you will be able to decide what action to take.
Example: Investigating Excessive Elapsed Time

This data center problem demonstrates that the biggest bottleneck is not always the cause of an elapsed time exception.

The following pages show how to use the OMEGAMON II for MVS historical trends and details panels to discover what is really interfering with the normal execution of a batch job.

Discovering the problem

You are monitoring your system from the System Status panel, and the Batch light turns yellow.

<table>
<thead>
<tr>
<th>Workload status</th>
<th>Resource status</th>
<th>Operator Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>S Batch  *****</td>
<td>_ CPU       -----</td>
<td>_ Key Task  -----</td>
</tr>
<tr>
<td>_ STC/APPC</td>
<td>_ DASD   *****</td>
<td>_ WTO Buffer  -----</td>
</tr>
<tr>
<td>_ TSO: RTA</td>
<td>_ Tape      -----</td>
<td>_ WTORs       -----</td>
</tr>
<tr>
<td>_ TSO Host</td>
<td>_ Paging     -----</td>
<td>_ OLTEP       -----</td>
</tr>
<tr>
<td>_ All P.G.</td>
<td>_ Storage    -----</td>
<td>_ DDR Swap    -----</td>
</tr>
<tr>
<td>_ Domains</td>
<td>_ CSA       -----</td>
<td>_ Max. Tasks  -----</td>
</tr>
<tr>
<td></td>
<td>_ Channels   -----</td>
<td>_ HSM        -----</td>
</tr>
</tbody>
</table>
On this panel, you see that job PIVJIT25 has an Elapsed time of 35:56 minutes and is the only job highlighted in yellow.

Now you know that the yellow Batch light on the System Status panel was caused by excessive elapsed time for job PIVJIT25. That is, the elapsed time for PIVJIT25 exceeds the elapsed time warning threshold of the job’s performance group.

To find out why the elapsed time is so high for this job, you enter s in the PIVJIT25 input field to reach the Details for a Job or Started Task panel shown in the following figure.

**Investigating excessive elapsed time**

The Status area on the left side of the panel shows that the elapsed time for PIVJIT25 continues to increase; it has reached 36:10 minutes.

The Elapsed time Profile area on the right side of the panel shows that CPU wait is the main bottleneck (31 percent) and I/O wait is the secondary bottleneck (20 percent). Your suspicion is that CPU wait caused the excessive elapsed time for PIVJIT25.
Before yielding to the temptation to reduce CPU wait time by moving PIVJIT25 to a performance group with a higher dispatching priority, you decide to find out if a high CPU wait percentage is normal for this job.

If high CPU wait is normal for this job, then the I/O wait must be causing the excessive elapsed time, even though that percentage is lower. If high CPU wait is not normal, then it is the cause of the excessive elapsed time. The historical trends panel can provide this information.

To reach the historical trends panel for this job, you:

- Select Historical Trends for a Batch Job or TSO User from the GoTo pull-down menu.
- When prompted for a date and time period, you enter 3 to request this hour for the previous days of this week.

In this example, you only need data for the previous two or three executions of this daily batch job.
The trends panel for PIVJIT25 shows that CPU wait was also the major wait reason for the previous three executions of the job, though the elapsed times were all less than 30 minutes.

You want to compare the CPU wait percent of one of the previous executions displayed on this panel with the CPU wait percent of the current job, which was 31 on the previous panel. If the CPU wait percent of the current job is significantly higher than one of the previous executions, then CPU wait can still be the cause of the excessive elapsed time of the current job.

You enter an s in the input field of one of the previous time periods to display historical details for that execution of the job.

The Bottlenecks area on the right side of this panel shows that the CPU wait percent (33.46) of yesterday’s successful execution of PIVJIT25 was higher than the the CPU wait percent of the current execution (31). However, despite the higher CPU wait percent, the elapsed time was only 28.55 minutes for yesterday’s execution.

Clearly, the seemingly high CPU wait for the current execution is not the cause of its excessive elapsed time.
Now that you know a high CPU wait is normal for this job, you look to see if any other waits are normal. No other wait reasons appear in the Bottleneck data for yesterday’s successful execution.

**The real cause of today’s problem**

You have discovered that the biggest bottleneck that appeared earlier today on the Details for a Job or Started Task panel (31 percent) was a normal occurrence for this job, even during those executions when it met service-level expectations.

The I/O wait of 20 percent that appeared on the same panel, is the real cause of the excessive elapsed time.
Example: Resolving an I/O Wait Problem

The following pages demonstrate how to use the Seek Analysis panel to discover what is interfering with the normal execution of a batch job.

Discovering the problem

You are monitoring your system from the System Status panel, and the Batch light turns yellow.

To request more information for this indicator, you issue the Analyze Problems action for Batch. To do this, position your cursor next to Batch, type a, and press Enter.

On this panel, you see that job PIVJIT25 has an Elapsed time of one hour fifteen minutes and is the only job highlighted in yellow.

Now you know that the yellow Batch light on the System Status panel was caused by an excessive elapsed time for job PIVJIT25. That is, the elapsed time for PIVJIT25 exceeds the elapsed time warning threshold of the job’s performance group.
To find out why the Elapsed Time is so high for this job, you enter s in the PIVJIT25 input field to reach the Details for a Job or Started Task panel shown in the following figure.

**Investigating an I/O wait problem**

The Elapsed Time Profile area on the right side of the panel shows that I/O wait is the main bottleneck (31 percent) and CPU wait is the secondary bottleneck (20 percent). Your first suspicion is that I/O wait caused the excessive elapsed time for PIVJIT25.

You enter s in the I/O wait execution state input field to display the Bottlenecks panel as shown in the following figure.
The Contention by Resource area on the right side of the panel shows that there are other jobs causing the delay.

You now want to investigate which other workloads are in contention for volume PRI003. You enter s in the Disk PRI003 queued state input field. This action gives you the DASD Details for a Device panel, as shown in the following figure.

The Response Time area on the right side of the panel shows a high disconnect time of 33.3 ms which may indicate Seek delays.
To investigate possible Seek delays place the cursor on the (Seek Analysis) push button and press Enter. The DASD Seek Analysis for a Device panel appears, as shown in the following figure.

This DASD Seek Analysis for a Device panel graphically shows you the:

- head movement on volume PRI003
- contention for the head exists between workloads PIVJIT25 and TOMTST12

Such contention creates Seek delays which help to account for the unusually high Queued I/O percentage displayed in the Bottlenecks panel.
You now want to place TOMTST12 into a performance group with an MPL of (0,0). To do this, enter fast path iwda to access the All Domains panel where you can see which performance group possesses the required characteristics of a minimum MPL of 0 and a maximum MPL of 0. The All Domains panel is shown in the following figure.

**Note:** Save the current jobname, TOMTST12, so that you can return the job to the same performance group when you are ready to resume execution.

**Solving an I/O wait problem**

On this panel, you see that Domain 111 has a minimum MPL of 0 and a maximum MPL of 0.
You now want to check the current IPS to find out which performance group domain 111 belongs to. To do this, enter fast path `icse` to display the System Environment panel, as shown in the following figure.

On this panel, you see that the system parameter is IPSGG. Browse the IEAIPSGG member of SYS1.PARMLIB. You see that domain 111 belongs to performance group 13. Now you are ready to reset the performance group of TOMTST12.
Enter fast path iwbd to reach the Details for a Batch Job or Started Task pop-up window, as shown below.

```
<CPU Utilization>   (System Environment)
F1=Help    F2=Keys    F3=Exit    F5=Refresh    F6=Console    F10=Action Bar
F11=Print     F15=System Status
```

Enter TOMTST12 in the Name of batch job or started task input field and press Enter. The Details for a Job or Started Task panel is displayed, as shown in the following

**Note:** Save the current performance group of TOMTST12, so that you can return the job to the same performance group when you are ready to resume execution.

Enter fast path ar to reach the Reset Performance Group pop-up window. Enter 13 in the Performance Group Number input field.
When job PIVJIT25 has completed you can reset the saved performance group of TOMTST12 so it can resume execution.
Example: Resolving a Batch Problem

The following pages show how to use the OMEGAMON II to investigate a problem in batch processing.

Discovering the problem

You are monitoring your system from the System Status panel, and you notice that the BATCH indicator is red.

To request more information for this indicator, you issue the Show Details action for BATCH. To do this, position your cursor next to BATCH, type s, and press Enter. This action displays the Batch Jobs Overview panel shown in the following figure.
Investigating the problem

You notice that job WORKWK1 has a long WAIT time. To show the details for this job, position the cursor on the input area, and type `s`. This will display the Details for a Job or Started Task panel as shown in the following figure.

Solving the problem

You notice that job WORKWK1 is waiting for a tape to be mounted. You can call Operations to ask them to mount the tape.
Example: Resolving a Started Task Problem

The following pages show how to use the OMEGAMON II Started Tasks Overview panel to discover what is interfering with the normal execution of jobs.

Discovering the problem

You are monitoring your system from the System Status panel, and the STC/APPC status light turns red.

To request more information for this indicator, issue the show details action for STC/APPC. To do this, position the cursor next to STC/APPC, type s, and press Enter. This will display the Started Tasks Overview panel as shown in the following figure.
**Investigating the problem**

You notice that job TSL2S20 is showing a high I/O rate and high CPU utilization. For more detailed information, position the cursor in the input area in front of the desired job, and type `/`. Return the cursor to the actions input area at the top of the panel, and type `v` to activate the pull-down action menu. Select the option that allows you to view the started tasks sorted by I/O rate. Investigate the job with the highest I/O rate to determine if DDNAME contention is causing the problem. Position the cursor in the input area of the desired job, type `b`, and press Enter. This will display the Bottlenecks panel showing active or queued I/O as shown in the following figure.
You notice that job TSL2S20 is waiting for CPU processing and PRI041 on device number 0280. For detailed information, position the cursor in the input area in front of the device number, and type / . This will cause the actions pull-down menu to appear as shown in the following figure.
Monitoring Workloads, Resources, and Alerts:

To display more detailed information, select the DDNAME activity in the action pull-down menu. Position the cursor at the input field for the action pull-down menu, and type **d** for DDNAME activity.

<table>
<thead>
<tr>
<th>KM2</th>
<th><em>D</em></th>
<th>1. Show details</th>
<th>necks</th>
<th>System: SP11 AUTO(60)</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>3. Historical details</td>
<td>x jobname or Service Class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jo</td>
<td>4. DDname activity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Im</td>
<td>5. Exit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Se</td>
<td>F1=Help F12+Cancel</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Productive | 100.0 | Using CPU | 6.9> |
| System     | .0    | Waiting for CPU | 46.6 |
| Disk PRI041 0280 Act | 46.6 |

The address space dataset allocations pop-up panel will be displayed as shown in the following figure.
Solving the problem

To display more detailed information, position the cursor next to the desired dataset name in the pop-up panel, and type `xph.s:exph.` This will cause the Dataset Details pop-up panel to appear as shown in the following figure.
The Dataset Details pop-up panel does not indicate a clear cause for contention in this example. The problem may be caused by other LPARs sharing access to device number 0280. You have to repeat this example on one or more of the other LPARs to see if they are using device number 0280 and causing the problem. It may be necessary to move some of the datasets to a different device, or restrict access to device number 0280 to eliminate the cross LPAR contention.
Example: Investigating a CSA Problem

The CSA status indicator is found in the second column of the System Status panel, under the heading "Resource status." When you spot a problem with common storage (a red alert CSA status indicator), you can navigate to the Analyze CSA Problems panel for more details. From there, you can navigate to the Common Storage Utilization panel where you can identify which area of CSA is having trouble, show details on that problem area, and see how much common storage is being used, and by whom.

Discovering the problem

Suppose you are monitoring your system from the System Status panel and you notice that the CSA indicator is red.

To request more information for this indicator, you issue the Analyze Problems action for CSA. To do this, position your cursor next to CSA, type A, and press Enter. This action displays the Analyze CSA Problems panel shown in the following figure.

On the following panel, you see that the percentage of CSA utilization has exceeded its thresholds. The CSA critical threshold is 95% and the current CSA in use is 97%.
Investigating a CSA problem

This panel lists four important areas of common storage (CSA, SQA, ECSA, ESQA). For each area, the panel provides the following information:

- amount of storage currently in use
- percentage of storage currently in use
- total size of each area, as specified at the initial program load (IPL) in member IEASYSxx of SYS1.PARMLIB
- amount of allocated but unowned storage (storage that was not freed when an address space terminated)
- growth in use since IPL

Areas that are highlighted on this panel are reaching dangerously high levels of allocation. You see that CSA utilization is 97%.

To find out why CSA has exceeded its thresholds, press Enter in the CSA utilization input field to reach the Common Storage Utilization panel shown in the following panel.
You request the **Show details** action for CSA. To do this, position the cursor next to CSA and press Enter. The Active Users of CSA panel appears.
This panel provides information on the users of CSA. You notice that task TESTIMS has 20% of the total CSA allocated.

You call the IMS support group and verify that this is a test IMS and there are no application developers currently using it. Therefore it can be cancelled to avoid system outage.

To cancel the test system, you type a slash (/) next to TESTIMS and press Enter. The Actions pull-down menu appears. You enter to cancel the task. The Confirm Request pop-up menu appears.

From the pop-up menu you select Process Request. TESTIMS terminates.

You go back to the Common Storage Utilization panel and press F5 to verify that CSA in use is less than before.
Example: Looking Up Cache Statistics

Because cache memory is used to reduce access time and therefore increase performance, OMEGAMON II provides a number of statistics regarding cache memory that help you fine tune your system’s performance.

OMEGAMON II reports cache statistics for cache controllers including models 3880-13, 3880-23, and 3990/2105. These statistics include cache read and write hit percentages and I/O requests indirectly related to cache.

The panels you need

The information presented on the following pages shows how to access the OMEGAMON II panels that allow you to control the monitoring of cache memory and display cache statistics.
Setting DASD exception thresholds

You can set thresholds relating to cache on the Set DASD Exception Thresholds pop-up, which includes four thresholds related to cache.

To access the Set DASD Exception Thresholds pop-up, follow this two-step procedure:
1. From any OMEGAMON II panel, enter fast path `otd` to reach the Specify DASD Groups pop-up.
2. From this pop-up, select the DASD threshold group you want to change.

You can set thresholds relating to cache on the Set DASD Exception Thresholds pop-up, which includes four thresholds related to cache.
Setting cache data collection frequency

The Control DASD Display pop-up includes a field to control the frequency of cache data collection.

To display this pop-up, enter l (lowercase L) on the input field before the DASD indicator on the System Status panel.

Analyzing DASD problems

The Analyze DASD Problems panel reports volumes that have exceeded the thresholds set for cache.

To display this panel, enter a (Analyze Problems) on the input field of the DASD status indicator on the System Status panel.
Finding inactive cache volumes

The DASD Response and % Busy panel reports the number of volumes that are eligible for cache but do not have cache activated.

To display this panel, enter **s** (Show Details) on the input field of the DASD status indicator on the System Status panel.

---

Finding read and write hit percentages

The DASD Details for a Device panel displays current cache read and write hit percentages and is accessed as follows:

- Enter **s** (Show Details) on the input field of the DASD indicator on the System Status panel.
- From the DASD Response and % Busy panel, select a device by typing **s** in the input field.
Monitoring Workloads, Resources, and Alerts:

Viewing realtime cache statistics

Realtime cache statistics are reported separately for 3880-13/23 and 3990/2105 DASD cache controllers.

- To access the panel showing 3880 statistics, enter fast path `iihc` from any OMEGAMON II panel.
- To access the panel showing 3990/2105 statistics, enter fast path `iihs` from any OMEGAMON II panel.

This is a sample panel for a 3990-3 device.
Viewing historical data for cache controllers

Historical details are available for the 3880-13/23 and 3990/2105 DASD cache controllers.

To display historical details for 3880-13 and 3880-23 cache controllers, enter fast path `iihh` from any OMEGAMON II panel.

To display historical details for 3990/2105 cache controllers, enter fast path `iihi` from any OMEGAMON II panel.

Viewing historical cache statistics

Historical trends for cache statistics are reported similarly to the historical details for controllers described above.

To access historical trends for cache statistics, enter fast path `iiht` from any OMEGAMON II panel.

Example: Replacing Link List Datasets

System administrators or other authorized users may want to update/replace link list datasets dynamically without performing an IPL. This example explains the process.

Note: This example applies only to users of OS/390™. Release 2 and below.

Use with CAUTION!

The utility invoked by the Replace Link List panel dynamically changes the system link list. The utility does not consider the tasks currently running on the system. To prevent library
inconsistencies and possible failure of tasks that are running, perform the replacement with caution.

**Authorizations**

To perform a link list replacement you must be authorized to issue an OMEGAMON QLLA command. By default the security for this command is level three.

If you select the option to update the LLA directory, OMEGAMON II must be:
- authorized to issue the MVS START and STOP commands.
- given READ authority to SYS1.PARMLIB.

**Before you begin**

You will have to modify members IEASYSxx, LNKLSTxx, and/or CSVLLAxx in SYS1.PARMLIB before you begin a link list dataset replacement.

Refer to *MVS/ESA Initialization and Tuning Guide* and the *MVS/ESA Initialization and Tuning Reference* for more information.
**Viewing all link list datasets**

The Link List Datasets panel presents a list of datasets with their search orders and APF-authorization statuses, as shown below.

To access the Link List Datasets panel, enter fast path `iksd` from any OMEGAMON II panel.

---

### Replacing a link list dataset

Replacing a link list dataset is a two-step procedure that ensures that no datasets will be dropped at the next IPL.

First you enter the suffix of the IEASYSxx member of SYS1.PARMLIB that will be used as input to the link list replace utility.

To enter the suffix, enter fast path `iksr` to reach the Replace Link List Dataset pop-up window which prompts you for the two-character suffix.

Ensure that the IEASYSxx member exists before proceeding.
After entering the suffix on the Replace Link List Datasets pop-up, the Replace Link List panel appears. This is where you perform the second step in the procedure, entering the request to replace the current link list.

To enter your request to replace the current link list with the proposed changes displayed on the panel, enter \texttt{r} (for Replace) in the Library Link List field.

If the Library Lookaside Area (LLA) facility is installed at your site, a pop-up appears that allows you to request the MVS STOP and START commands required to update the LLA directory with the new link list datasets.
Updates to the link list will take place after you confirm your request on a confirmation pop-up window.

<table>
<thead>
<tr>
<th>Status</th>
<th>Volume</th>
<th>APF</th>
<th>Ord</th>
<th>Dataset Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>MP430G</td>
<td>Yes</td>
<td>17</td>
<td>SYS1.SCBDDENU</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>1</td>
<td>SYS1.LINKLIB</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>2</td>
<td>SYS1.MIGLIB</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>3</td>
<td>SYS1.CSSLIB</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>4</td>
<td>SYS1.CMCLIB</td>
</tr>
<tr>
<td>Old</td>
<td>CAN004</td>
<td>Yes</td>
<td>5</td>
<td>CAN.SYSSG.LINKLIB</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>6</td>
<td>SYS1.DGTLIB</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>7</td>
<td>SYS1.DFO.V2R6M0.DFQLLIB</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>8</td>
<td>SYS1.ICE.V1R1M1.SORTLIB</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>9</td>
<td>SYS1.ISPF.V3R3.LINKLIB</td>
</tr>
<tr>
<td>Old</td>
<td>CAN001</td>
<td>Yes</td>
<td>10</td>
<td>CAN.UTILITY</td>
</tr>
<tr>
<td>Old</td>
<td>CAN001</td>
<td>Yes</td>
<td>11</td>
<td>CAN.MVSX.LINKLIB</td>
</tr>
<tr>
<td>Old</td>
<td>PPSMPA</td>
<td>Yes</td>
<td>12</td>
<td>CAN.TCPIP.V2R2M1.SEZALINK</td>
</tr>
<tr>
<td>Old</td>
<td>CAN001</td>
<td>Yes</td>
<td>13</td>
<td>CAN.PP.LINKLIB</td>
</tr>
<tr>
<td>Old</td>
<td>PPSMP7</td>
<td>Yes</td>
<td>14</td>
<td>PP.CAICA1.R50.CAILIB</td>
</tr>
<tr>
<td>Old</td>
<td>CAN003</td>
<td>Yes</td>
<td>15</td>
<td>SYS1.COBLIB</td>
</tr>
<tr>
<td>Old</td>
<td>CAN004</td>
<td>Yes</td>
<td>16</td>
<td>SYS1.PL1.NEWLINK</td>
</tr>
<tr>
<td>Old</td>
<td>MP430G</td>
<td>Yes</td>
<td>18</td>
<td>SYS1.NUCLEUS</td>
</tr>
</tbody>
</table>
Example: Resolving a Channel I/O Problem

The following pages demonstrate how to use the OMEGAMON II Channel Activity panel to investigate a problem in the system's job flow.

Discovering the problem

You are monitoring your system from the System Status panel, and the Channels status light turns red.

To request more information for this indicator, issue the show details action for channels. To do this, position the cursor next to Channels, s, and press Enter.

This will display the Channel Activity panel as shown in the following figure.
Investigating the problem

On this panel, you see that CHPID "84" is showing the highest activity. To obtain more information, enter "/" in the CHPID "84" input field.

This will display the Actions pull-down menu as shown in the following figure.

To determine the type of device attached to the CHPID, select option 1 from the pull-down menu.
If both TAPE and DASD devices are attached to the same CHPID, a pop-up will appear allowing selection of either device type. The Channel Activity panel with a pop-up will appear as shown in the following figure.
In this example, option 2 is selected. This will cause the Tape Drives panel to appear as shown in the following example.

To display device information, position the cursor on the input area in front of the desired device, type `s`, and press enter.

This will cause the Tape Unit Details panel to appear as shown in the following figure.

**Solving the problem**

In this example, the tape drive dropped ready. You could call Operations and have them ready the tape drive.
Investigating the problem

If you selected Option 1 (Display DASD attached to the CHPID) on the Channel Activity pop-up panel, then the DASD Response and % Busy panel would appear as shown in the following figure.
You notice the high response time for device number 9C3 of 255.3 ms. Enter 's' in the input field in front of device number 9C3. This will display the DASD Details for a Device panel as shown in the following figure.
### Solving the problem

<table>
<thead>
<tr>
<th>KM2D02D</th>
<th>DASD Details for a Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume:</td>
<td>DevNo . . : 09C3 LCU . : 012</td>
</tr>
<tr>
<td>Mount :</td>
<td>privat Status . . : Online</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>Current Status</td>
<td>Response Time Over Last 21:57 MN</td>
</tr>
<tr>
<td>Free space .</td>
<td>00533.00005</td>
</tr>
<tr>
<td>Largest Block</td>
<td>00533.00000</td>
</tr>
<tr>
<td># Open DCBs</td>
<td>0</td>
</tr>
<tr>
<td>% BUSY . .</td>
<td>0</td>
</tr>
<tr>
<td>I/O Rate . .</td>
<td>0 / sec</td>
</tr>
<tr>
<td>I/O Queue. .</td>
<td>0</td>
</tr>
<tr>
<td>Dev Busy Delay</td>
<td>0 ms</td>
</tr>
<tr>
<td>CU Busy Delay</td>
<td>0 ms</td>
</tr>
<tr>
<td>Dir Pt Busy dl</td>
<td>0 ms</td>
</tr>
<tr>
<td>Cache (R/W)</td>
<td>74.8%/100%, 0%</td>
</tr>
<tr>
<td>-----------------+-----------------</td>
<td></td>
</tr>
<tr>
<td>IOS Queue</td>
<td>245.4 ms</td>
</tr>
<tr>
<td>Pending</td>
<td>.2 ms</td>
</tr>
<tr>
<td>Connect</td>
<td>3.5 ms</td>
</tr>
<tr>
<td>Disconnect</td>
<td>4.2 ms</td>
</tr>
<tr>
<td>Total</td>
<td>255.3 ms</td>
</tr>
</tbody>
</table>

The high IOS queue response time shows that there is a scheduling conflict for this device. This could be resolved by implementing I/O priority queueing or moving the datasets causing the scheduling conflict.
Example: Resolving a CPU Utilization Problem

The following pages show how to use the OMEGAMON II to investigate a problem with CPU utilization.

Discovering the problem

You are monitoring your system from the System Status panel, and you notice that the CPU indicator is red.

To request more information for this indicator, you issue the Show Details action for CPU. To do this, position your cursor next to CPU, type s, and press Enter. This action displays the CPU Utilization panel shown in the following figure.
Monitoring Workloads, Resources, and Alerts:

Investigating the problem

The high CPU % and high TCB % shown above would indicate that test job TDM2S4E is in a LOOP condition.

Solving the problem

Cancel the job, and system performance will return to normal.
Monitoring Workloads, Resources, and Alerts:

Example: Investigating a Key Task Alert

The Operator Alerts section of the System Status panel provides information especially for operators of MVS systems as well as providing useful information for all OMEGAMON II users.

For example, the Key Task status indicator on the System Status panel gives the status of all critical started tasks on your system. The light for Key Task becomes red or yellow if a critical started task or job stopped running unexpectedly or did not start at all.

Example of a key task status alert

Suppose, for example, that Key Task has a red status alert indicating a critical job or started task has a problem. (In the panel shown below, $$$$$ indicates a red status light.) To find the reason for the alert, request the Show Details action for this object by typing S in the Key Task input field and pressing Enter.

<table>
<thead>
<tr>
<th>Workload status</th>
<th>Resource status</th>
<th>Operator Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Batch</em></td>
<td><em>CPU</em></td>
<td><em>Key Task $$$$$</em></td>
</tr>
<tr>
<td><em>STC/APPC $$$$$</em></td>
<td><em>DASD $$$$$</em></td>
<td><em>WTO Buffer -----</em></td>
</tr>
<tr>
<td><em>TSO: RTA</em></td>
<td><em>Tape</em></td>
<td><em>WTORs $$$$$</em></td>
</tr>
<tr>
<td><em>TSO Host</em></td>
<td><em>Paging *****</em></td>
<td><em>OLTEP</em></td>
</tr>
<tr>
<td><em>All P.G.</em></td>
<td><em>Storage $$$$$</em></td>
<td><em>DDR Swap</em></td>
</tr>
<tr>
<td><em>Domains</em></td>
<td><em>CSA $$$$$</em></td>
<td><em>Max. Tasks</em></td>
</tr>
<tr>
<td><em>Channels *****</em></td>
<td><em>HSM</em></td>
<td><em>Key DASD *****</em></td>
</tr>
</tbody>
</table>

F1=Help  F2=Keys  F3=Exit  F5=Refresh  F6=Console  F10=Action Bar  F11=Print
Key Task Alert window

The Key Task Alert pop-up window is displayed as shown below. It shows which critical tasks are missing.

<table>
<thead>
<tr>
<th>Workload status</th>
<th>Resource status</th>
<th>Operator Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ Batch _</td>
<td>_ CPU _</td>
<td>s Key Task $$$$ _ Enqueue _</td>
</tr>
<tr>
<td>_ STC/APPC $$$$</td>
<td>_ DASD _</td>
<td>Key Tasks Alert</td>
</tr>
<tr>
<td>_ TSO: RTA _</td>
<td>_ Tape _</td>
<td>The following critical tasks are missing: CICS IMS TSO VTAM</td>
</tr>
<tr>
<td>_ TSO Host _</td>
<td>_ Paging _</td>
<td></td>
</tr>
<tr>
<td>_ All P.G. _</td>
<td>_ Storage _</td>
<td></td>
</tr>
<tr>
<td>_ Domains _</td>
<td>_ CSA _</td>
<td></td>
</tr>
<tr>
<td>_ Channels _</td>
<td>Press Enter to go to console.</td>
<td></td>
</tr>
</tbody>
</table>

To restart a task or to see why it stopped, you must go to the system console. However, to go to the system console, you must have authorization.

**Note:** To obtain authorization to go to the system console, see your OMEGAMON II Customizer.

Upon obtaining authorization to go to the system console, press Enter from the Key Tasks Alert pop-up window. The system console is displayed.
Example: Responding to an Enqueue Conflict

As another example of an operator alert, we present an enqueue conflict. In this example, the Operator’s Toolkit, a special feature of OMEGAMON II, is introduced as a tool for handling an enqueue problem or other operator alert problems.

Example of an enqueue conflict

Suppose that a user has detected a “deadly embrace” because another job is vying for use of the same dataset and the user is not authorized to cancel the job. The user then calls the system operator.

The operator selects Toolkit from the Index pull-down, then Operator’s Toolkit.

(The operator could also have used the fast path ik.)

To determine where the conflict is, the operator selects 3, Enqueue Conflicts and Reserves from the Operator’s Toolkit window as shown below.

```
<table>
<thead>
<tr>
<th>Workload status</th>
<th>Select one of the following, then press Enter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>1. DASD Response and Percent Busy</td>
</tr>
<tr>
<td>STC/APPC</td>
<td>2. Enqueue and Reserves Details</td>
</tr>
<tr>
<td>TSO: RTA</td>
<td>3. Enqueue Conflicts and Reserves</td>
</tr>
<tr>
<td>TSO Host</td>
<td>4. SMF Dataset Information</td>
</tr>
<tr>
<td>All P.G.</td>
<td>5. SMF Subsystem Information</td>
</tr>
<tr>
<td>Domains</td>
<td>6. System Console</td>
</tr>
<tr>
<td>Channels</td>
<td>7. Tape Details for a Drive</td>
</tr>
<tr>
<td>HSM</td>
<td>8. Tape Drives</td>
</tr>
</tbody>
</table>
```

F1=Help  F2=Keys  F3=Exit  F5=Refresh  F6=Console  F10=Action Bar  F11=Print
The Enqueue Conflicts and Reserves panel appears, listing the enqueue conflicts currently existing. On this panel, the operator types / (slash) before the job to be cancelled.

**Warning**: Be extremely careful when cancelling a job.

Once a job is selected, the Actions pull-down menu is displayed as shown below. The operator selects 6 to cancel the job.
Toolkit for system programmers

Note: There is also a Toolkit for System Programmers in OMEGAMON II. It contains special system programmer functions that are not available through status alert lights. Like the Operator’s Toolkit, the System Programmer’s Toolkit is accessed from the Index pull-down; the fast path is iks.
Monitoring Performance across Systems

Previous sections of this chapter described how to monitor the performance of a single system using OMEGAMON II panels. This section describes how to monitor more than one system at a time by:

- displaying realtime information from two or more systems on the same display screen using classic OMEGAMON commands
- displaying historical information about a shared-DASD device in a multi-system environment using classic EPILOG commands

In this section

In this section we cover monitoring realtime performance across systems.

Displaying information from multiple systems

You can display realtime information from two or more systems on the same display screen through the Cross Memory (XMF) and Cross System (XSF) facilities.

These facilities enable you to enter commands for, and receive information from, the following Candle products on one OMEGAMON display screen:

- OMEGAMON for MVS
- OMEGAMON for CICS
- OMEGAMON II for DB2® (XMF only)
- OMEGAMON for IMS
- OMEGAMON for VM (XSF only)

How is this accomplished?

One of the products listed above, typically OMEGAMON for MVS, acts as a director running in either dedicated or VTAM mode. The director controls the display screen and communicates with the other OMEGAMON products by passing your commands to the appropriate product acting as a collector. The collector passes the output of the commands back to the director, which in turn displays the output on its display screen.

For the sake of simplicity, the examples presented here will focus on using an OMEGAMON for MVS director in dedicated mode.
Cross memory/cross system modes

A single director can communicate with up to seven OMEGAMON collectors. You have a choice of mode for each collector:

cross memory mode

Collects information from OMEGAMON products that are running on the same MVS system. This is useful if you want to limit the number of dedicated terminals you have or provide a single point where you can monitor MVS, IMS, and CICS regions.

In this mode, the director communicates with the collectors through a window in the Common Service Area (CSA).

cross system mode

Collects information from OMEGAMON products that are running on entirely different MVS operating systems, even if those systems are running at different levels of MVS.

In this mode, the director communicates with the collectors through a dataset that is accessible by both.

A director can communicate with collectors in both modes at the same time and on the same director display screen, enabling you to pull together OMEGAMON displays from all over your site.

Preparing to use XMF and XSF

Before you begin using the cross memory and cross system facilities, you must start a director and the desired OMEGAMON collectors.

Check with your OMEGAMON II Customizer to see if:

- the cross system and cross memory facilities have been installed according to the instructions provided in the OMEGAMON II for MVS Configuration and Customization Guide.
- a dedicated 327x terminal is available
- the appropriate director and collectors have been started

Once started, collectors running in cross memory mode automatically appear on your display terminal. If you plan to use a collector running in (one that is gathering information from an OMEGAMON on another MVS system), you must first use the ATTACH INFO-line command to identify to the director which dataset is to be used for cross system communication; information gathered by this collector will not appear on your display terminal until this step is performed.
Displaying XMF/XSF information on a screen

When the XMF/XSF director and collectors have been started, the screen display on the dedicated 327x terminal divides into segments to reflect activity for the director and each collector simultaneously. The director segment appears first, followed by segments for up to seven collectors.

An INFO-line shows where each segment begins. You can type OMEGAMON commands on each of the INFO-lines; when you press Enter, the commands are executed by the respective collectors and their results displayed in the appropriate segments.

The following is an example of an XMF/XSF screen reflecting an OMEGAMON for MVS and OMEGAMON for CICS session running on CPU A033, and an OMEGAMON for IMS session running on CPU IPRD.

```
#01       DIR      OM/DEX   V&RELO../C A033  date. 16:58:45
...
OMEGAMON/MVS commands (running on A033)
...
#01       XMM      OM/CICS  V520.00 CPRD   date. 16:58:45
...
&OC. commands (running in the CICS production region)
...
#01       DSK      OM/IMS   V520.99 IPRD   date. 16:58:45
...
OMEGAMON/IMS commands (running on IPRD)
...
```

In this example, the screen contains three INFO-lines that identify the start of a screen space dedicated to each of the three monitors. Each segment is identified by the 4-character system ID, which immediately precedes the date. In this case, the first OMEGAMON segment is identified by A033 (which represents the SMF ID for this system), the OMEGAMON for CICS segment is identified by CPRD, and the OMEGAMON for IMS segment by the ID IPRD.

Each segment also has a 3-character field to the right of the screen space identifier that denotes which type of segment it is:

- **DIR**: The director segment
- **XMM**: A cross memory mode segment
- **DSK**: A cross system mode segment
Controlling the XMF/XSF screen

Each cross memory/cross system segment acts like a normal OMEGAMON screen. You can perform any of the following actions within a segment, independently from the other segments:

- enter OMEGAMON commands
- scroll up and down
- use function keys and ring the terminal bell (note, however, that in this case there is no indication as to which segment was responsible)

You can further control the XMF/XSF screen as follows:

- When displaying information from an OMEGAMON that supports extended color and highlighting capabilities, the collector recognizes only the first color encountered for each line.
- If you press the CLEAR or PA1 key, an attention is propagated to each collector segment in a process that may take one or two cycles to complete. (This is useful if you accidentally turn on extended color for a terminal that is not equipped for extended color support, and a reset is required.)
- You can control the number of lines displayed for each collector segment by using the /GIVE and /TAKE INFO-line commands to transfer lines from one segment to another. See the OMEGAMON for MVS Command Language Reference Manual for information on these commands.

Synchronizing cross memory collectors

In cross system mode, the collectors run on a different system from the director, so they communicate with each other through a dataset which resides on a DASD volume shared by both systems. This dataset acts as a temporary repository for commands and data, thus freeing the director from waiting for the collectors for an answer.

In cross memory mode, on the other hand, a director communicates with a collector through a window in the MVS Common Service Area (CSA), and must wait for a response from the cross memory collector. The director attempts to synchronize with each cross memory collector so that each collector responds immediately to new commands that you issue; any interval-setting commands such as .SET INTERVAL are ignored. The director waits for each collector only for as long as the director’s cycle time (five seconds by default). If a collector fails to respond in this interval, the director stops waiting for that collector and goes on to the others.

When a collector falls out of sync, any commands you enter within its segment take one cycle to process, instead of being handled immediately. You will also notice a time difference between the time displayed on the collector’s INFO-line and that of the director.
If a collector can respond within the director’s cycle time but is very slow, it causes the director to slow down as well. You might find this condition with a low-priority CICS or IMS test region. You can eliminate this problem by using the /WAIT OFF INFO-line command in the problem segments to tell the director not to wait for these slow collectors. To synchronize the director and the collector again at any time, use the /WAIT ON INFO-line command. See the OMEGAMON II for MVS Command Language Reference Manual for information on the /WAIT command.

Do directors act differently in VTAM mode?

This section assumes your OMEGAMON II Customizer has installed the cross memory and cross system facilities in dedicated mode. If XMF and XSF have been installed in VTAM mode, you will see the following differences:

- If you press a PF key while the screen is updating, your keyboard will lock. To correct this situation, press the RESET key.
- INFO-line commands do not process until you press Enter.
- The INTERVAL= parameter of the .SET command displays and changes the update interval.
- The .AUPOFF and /AUPOFF commands are ignored by the VTAM mode director.
Monitoring Historical Shared-DASD Performance

While monitoring your system, suppose you notice that workloads are often waiting for I/O to a particular device. You know that this device is shared by more than one system, and you would like to see how it has been used by workloads and performance groups on the other systems to help you decide how to alleviate the I/O problems on your system.

The following paragraphs describe how to use the JDAS and PDAS keywords of the EPILOG DISPLAY command to display historical information about a shared-DASD device in a multi-system environment.

Procedure

To show the cross-system impact of workload utilization against a particular shared-DASD device, use the following procedure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Make sure your OMEGAMON II Customizer has defined all systems to be monitored to the EPILOG reporter session.</td>
</tr>
<tr>
<td>2</td>
<td>Type <code>ge</code> in the action bar input field of an OMEGAMON II panel and press Enter to prepare to enter EPILOG commands.</td>
</tr>
<tr>
<td>3</td>
<td>Enter the DISPLAY command with the JDAS or PDAS keyword. Supply the volser and appropriate time period. See the <em>EPILOG for MVS Command Language Reference Manual</em> for information about the EPILOG JDAS and PDAS keywords.</td>
</tr>
<tr>
<td>4</td>
<td>When you have completed your shared-DASD investigation, type <code>end</code> on the EPILOG command line and press Enter to return to OMEGAMON II.</td>
</tr>
</tbody>
</table>
Example 1: shared-DASD degradation by workload

Suppose you have determined that there may be cross-system contention on volume COM002. You enter the following EPILOG DISPLAY command to display degradation by workload between 2:15 and 2:30 PM yesterday:

\[
\text{DISPLAY JDAS(COM002) YDAY STIME(14:15) ETIME(14:30)}
\]

The following report displays:

```
+==============================================================================+
<table>
<thead>
<tr>
<th>Volume = COM002                 Period = 14:15 to 14:30 on date.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Usage of selected DASD device for Cross-system ACTIVE I-O</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Workload_JES #<em>Sysid_Time</em>______%__</td>
</tr>
<tr>
<td>ASMPBKUP J4661 SYSA  5:32 M  82.1</td>
</tr>
<tr>
<td>$AOASMP1 S4533 SYSF  45:98 S  11.4</td>
</tr>
<tr>
<td>TDNY30 T4014 SYSA  26:22 S   6.5</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Relative Usage of selected DASD device for Cross-system QUEUED I-O</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Workload_JES #<em>Sysid_Time</em>______%__</td>
</tr>
<tr>
<td>$AOASMP1 S4243 SYSF  29:64 S  66.7</td>
</tr>
<tr>
<td>ASMPBKUP J8812 SYSA  13:94 S  31.4</td>
</tr>
<tr>
<td>$AOASMP2 S4533 SYSF :84 S  1.9</td>
</tr>
</tbody>
</table>
+==============================================================================+
```

In this figure, you can see both the delayed and the heavily active workloads accessing the volume on a comparative basis. For instance, job $AOASMP1 on SYSF was queued for I/O 66.7% of the time, while job ASMPBKUP on SYSA spent 82.1% of its time during the interval actively performing I/O.

The Active I/O and Queued I/O status percentages each add up to 100%. These percentages reflect the portion of the total observed active or queued I/O against volume COM002 that is attributable to each job. In other words, the number of samples for active I/O against COM002 during the time period 2:15 to 2:30 is accumulated from all eligible records on each datastore in the active datastore list. Each sample is weighted according to the sampling interval time in effect for each system involved.

In addition to the total accumulation, an accumulation for each job is also performed. The percentages are then calculated by dividing the job within system accumulations by the total accumulation across all jobs in all systems. This calculation is applied separately to active, queued, and reserved I/O wait reasons.
Example 2: shared-DASD degradation by performance group

If you change the shared-DASD keyword in Example 1 from JDAS to PDAS, you will see the impact of performance groups, rather than workloads, on COM002. For example, you enter the following command:

```
DISPLAY PDAS(COM002) YDAY STIME(14:15) ETIME(14:30)
```

The following report displays:

```
+==============================================================================+
<p>| Volume = COM002                  Period = 14:15 to 14:30 on date.           |</p>
<table>
<thead>
<tr>
<th>relative Usage of selected DASD device for Cross-system ACTIVE I-O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perf Grp_______Sysid_Time________%__</td>
</tr>
<tr>
<td>HOTBATCH       SYSA  5:32 M   82.1</td>
</tr>
<tr>
<td>REGBATCH       SYSF  45.98 S   11.4</td>
</tr>
<tr>
<td>PERF121        SYSA  26.22 S    6.5</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Relative Usage of selected DASD device for Cross-system QUEUED I-O</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Perf Grp_______Sysid_Time________%__</td>
</tr>
<tr>
<td>REGBATCH       SYSF  30.48 S   68.6</td>
</tr>
<tr>
<td>HOTBATCH       SYSA  13.94 S   31.4</td>
</tr>
</tbody>
</table>
|------------------------------------------------------------------------------+
```

In this figure, symbolic names are defined for each performance group in the Perf Grp column. If these names have not been defined, PERFnnnn. is displayed, where nnnn is the performance group number.

In this example, the jobs $AOASMP1 and $AOASMP2 have been defined in the same performance group, REGBATCH. The queued I/O portion of the display contains a rolled up percentage for REGBATCH for these two jobs, which appeared separately in the figure on page 135.

Recommendations

You can use the information provided in the JDAS and PDAS displays to help you decide how to alleviate I/O problems on a system. If you've issued the DISPLAY command with different time periods and you see a trend in DASD use among systems, you might take one of the following actions to alleviate device contention:

- allocate datasets needed by jobs on different systems to different devices
- spread workloads among systems more evenly
Introduction

When the applications that run on your system are not executing at peak efficiency, the resulting delay in service can set off a chain reaction of missed deadlines and reduced productivity.

This chapter describes how to analyze workload performance on your system by

- explaining bottleneck and impact analysis
- describing their associated controls and settings
- presenting two scenarios that show how to resolve typical bottlenecks
- providing general guidelines for handling common bottleneck situations

Chapter Contents

Understanding Bottleneck and Impact Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 138
Collecting Bottleneck and Impact Analysis Information . . . . . . . . . . . . . . . . . . . . . . . . 140
Excluding Wait Reasons from Bottleneck Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 143
Example: Diagnosing a Current Bottleneck . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145
Example: Analyzing a Recurring Bottleneck . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
Guidelines for Troubleshooting Bottlenecks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
OMEGAMON II surveys system resources to determine where your workloads are spending their time, and helps you see if they are spending that time productively or unproductively; it does this by reporting whether they are active or waiting for a resource. If a workload is waiting for a resource, OMEGAMON II also identifies the impact that other workloads may be having on the workload.

What is bottleneck analysis?
A bottleneck is a condition that causes a workload (a batch job, started task, or TSO session) to spend its time unproductively, preventing it from reaching its service goal. In helping you identify bottlenecks, OMEGAMON II breaks down the elapsed or response time of a workload into the following execution states. These execution states identify the current activity of a workload.

<table>
<thead>
<tr>
<th>IF the execution state is...</th>
<th>THEN the workload is...</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>actively using the CPU or performing I/O.</td>
</tr>
<tr>
<td>idle</td>
<td>voluntarily inactive. This includes swap conditions that the workload issues for itself when it expects to be waiting for a long period of time.</td>
</tr>
<tr>
<td>delayed</td>
<td>prevented from continuing its processing because it is waiting for a resource.</td>
</tr>
</tbody>
</table>

OMEGAMON II considers workloads to be in a bottleneck situation when their productivity is degraded because they are waiting for resources.

What causes a workload delay?
Wait reasons explain why the execution of a workload is delayed. Examples of wait reasons include:
- waiting for CPU
- swapped out to improve central storage availability

Can an active workload be constrained by a bottleneck?
A workload does not have to be waiting for a resource to be constrained by a bottleneck. Although a workload that is actively using the CPU or performing I/O is usually considered to be executing productively, its performance may be degraded if the amount of time it spends performing these activities is excessive due to the activity of another workload. Thus, even an active workload can be in a bottleneck situation.

From bottleneck analysis to impact analysis
Once OMEGAMON II determines the execution state of a workload, it then identifies other workloads or performance groups that are impacting the workload by contending for the same resources. Once you know the source of the resource contention, you can analyze the
contention between the workloads and take appropriate action, depending on their relative importance.

**Displaying bottleneck information in OMEGAMON II**

The OMEGAMON II Bottlenecks panel contains both bottleneck and impact analysis information. There are four ways to access this panel:

- Enter the `b` action code next to a problem workload.
- Select **Bottlenecks** from the Actions or GoTo pull-downs where available.
- Enter fast path `iwro` from any OMEGAMON II panel.
- Select **Resource Bottlenecks** from the Actions pull-down for a wait reason on the Details for a Job or Started Task panel. These detail panels list the execution states of the address space; you can press F1 on an execution state for recommendations on where to turn next.

**Resolving the bottleneck**

Once you have identified the cause of the bottleneck, you can resolve the situation in a number of ways. For guidelines on resolving bottlenecks, use the online help system on the Bottlenecks panel and read “Guidelines for Troubleshooting Bottlenecks” on page 154.

**Multi-tasking workloads**

Some workloads are capable of performing in multiple execution states simultaneously. An example is a workload that is actively using CPU while delayed waiting for a tape mount. Such workloads are considered multi-tasking and, therefore, capable of achieving resource total percentages greater than 100%.
Collecting Bottleneck and Impact Analysis Information

Introduction
By default, OMEGAMON II collects enough bottleneck and impact analysis information to solve short term, immediate problems. This section explains how to override this default collection interval if necessary.

Solving current problems
OMEGAMON II begins collecting bottleneck and impact analysis information automatically when you display the Bottlenecks panel, and (by default) stops collecting information when you leave the Bottlenecks panel. This default collection interval provides the information you need to resolve a current bottleneck problem.

Solving recurring problems
If bottleneck situations occur frequently during the day, or if the execution of a particular workload is frequently delayed, you may want to watch these situations over a longer period of time. The Session Defaults pop-up enables you to broaden the collection interval, activating bottleneck and impact analysis even while you’re using other panels, up to the duration of your OMEGAMON II session.

Implications of continuing collection
There is a significant amount of overhead associated with collecting bottleneck and impact analysis information. Therefore, we recommend that you keep the defaults of Off and No (for bottleneck and impact analysis, respectively) for resolving most short-term bottleneck problems, and change the defaults only when long-term bottleneck situations occur.

Session Defaults: fast path
The fast path for setting session defaults is ocs.

Session Defaults pop-up
The following figure is an example of the Session Defaults pop-up. We will limit our discussion of the session defaults you can specify to those related to bottleneck and impact analysis:

- bottleneck analysis
- enable impact analysis
- impact analysis time-out period
Collecting Bottleneck and Impact Analysis Information

Bottleneck and impact analysis defaults

The default entries on the Session Defaults pop-up for bottleneck and impact analysis are Off and No, respectively. As previously explained, this means that collection of bottleneck information will not begin until you display the Bottlenecks panel, and will end when you leave the Bottlenecks panel.

To troubleshoot long-term bottleneck problems, you must change these settings to On and Yes. This causes collection of bottleneck and impact analysis information to continue after you have exited the Bottlenecks panel, until you turn collection off or end your OMEGAMON II session.

You can turn collection off at any time during your OMEGAMON II session by returning to the Session Defaults pop-up and restoring the default entries for bottleneck (Off) and impact (No) analysis.
Impact analysis time-out period

Once you enable impact analysis, you can use the Impact analysis time-out period field to control how long OMEGAMON II will collect impact analysis information after you exit the Bottlenecks panel.

<table>
<thead>
<tr>
<th>TO continue for...</th>
<th>ENTER...</th>
</tr>
</thead>
<tbody>
<tr>
<td>a certain number of minutes</td>
<td>a number from 1 to 99.</td>
</tr>
<tr>
<td>the duration of your OMEGAMON II session</td>
<td>Off (time-out not in effect).</td>
</tr>
</tbody>
</table>
Excluding Wait Reasons from Bottleneck Analysis

Introduction
Some wait reasons may be of more interest to you than others when you are performing bottleneck analysis on a workload. OMEGAMON II enables you to specify the wait reasons that you would like to exclude from bottleneck analysis through a selection on the Options pull-down.

Excluding wait reasons: fast path
You select the wait reasons you want to exclude using the Wait Reason Reporting Control pop-up; the fast path is ocw.

Types of wait reasons to exclude
The Wait Reason Reporting Control pop-up enables you to exclude any of the following types of wait reasons:
- CPU waits
- enqueues
- idle waits
- paging waits
- SRM delays
- tape mounts
- other waits

OMEGAMON II always monitors and reports workloads that are actively using CPU, actively performing I/O, or queued to perform I/O. Therefore, you will not see selections for these execution states on the Wait Reason Reporting Control pop-up.

Rules for selecting wait reason types
By default, all wait reason types are selected. To exclude a wait reason type from being reported, erase the slash (/) next to the wait reason type.

If you exclude one or more wait reason types, then for the period of their exclusion, OMEGAMON II will confine its wait reason reporting to the remaining wait reason types. For instance, if you exclude tape mount delays and the actual amount of time the workload spent waiting for a tape mount was 20%, OMEGAMON II recomputes the other wait reason values to absorb that 20%.

Example: Wait Reason Reporting Control pop-up
The Wait Reason Reporting Control pop-up is shown below. In this example, all wait reasons have been selected for reporting except for enqueues and idle waits.
For more information

For a list of the specific wait reasons that are associated with each wait reason type, and for more information that may help you decide whether to exclude a wait reason type or not, press F1 next to the wait reason type.
Example: Diagnosing a Current Bottleneck

The following is a scenario that shows how you might use OMEGAMON II to resolve a current resource contention problem.

Selecting All P.G.

Suppose you are monitoring your system from the System Status panel when you notice that the All P.G. status indicator is red. First, you request the Analyze Problems action for All P.G.. To do this, position your cursor next to All P.G., type `a`, and press Enter.

OMEGAMON II displays the Analyze Performance Group Problems panel, as shown below. This panel displays all activity contributing to the critical condition of a performance group, and the current warning and critical threshold settings for each problem performance group.

In this example, the first period response times of TSO performance group numbers 98 and 15 have exceeded their critical thresholds.

Gaining perspective on other performance groups

To gain some perspective on how these performance groups are performing compared to other groups, you enter `s` next to one of the problem performance groups. OMEGAMON II displays the All Performance Groups panel, shown below. This panel summarizes each performance group, including those with problems.

In this example, performance group 98, PROD TSO, is experiencing the most difficulty in completing its work, as indicated by its low number of completed transactions and its high transaction times.
Choosing a performance group

Based on the following criteria, you decide to select PROD TSO for further investigation.

- It is a high-priority performance group.
- Its number of transactions is unusually low (343), well below expected service levels.
- Its elapsed time is unusually high, as reflected in the Average Transaction Times column.

You request the Bottlenecks action for PROD TSO. To do this, type `b` on the line next to PROD TSO and press Enter.

Identifying bottlenecks and impactors

OMEGAMON II displays the Bottlenecks panel. This panel gives you information about what other performance groups are impacting PROD TSO, as well as a list of its execution states and wait reasons.

The left side of the panel tells you that the BATCH performance group is the greatest impactor on PROD TSO, because for 20.1% of the time, it had resources needed by PROD TSO. The right side of the panel indicates resource contention; CPU wait accounts for 52.2% of the transaction time for PROD TSO.
Example: Diagnosing a Current Bottleneck

The wait reasons listed on the right side of this panel (such as CPU wait and swap page-in wait) have various best actions for their resolution. By using the online help system, you can display recommendations for how to handle each type of wait reason.

**Identifying greatest impactor**

Now you need to see detailed information about CPU wait, so you request the Show details action for CPU wait. To do this, position your cursor next to CPU wait and press Enter.

OMEGAMON II displays the Resource Bottleneck panel. This panel identifies which jobs or users, in which performance groups, are using the CPU.

As shown in this figure, PAYROLL is using 63.1% of the CPU. No wonder PROD TSO is suffering.
### Taking action

You open the Actions pull-down to review your options. To do this, type a slash (/) next to PAYROLL, then enter a in the input area of the action bar. OMEGAMON II displays the Actions pull-down for the Resource Bottleneck panel, as shown below.

```
___ Actions GoTo Index View Options Help
---------------------------------------------------------mm/dd/yy 11:14:12
XM2W09D Resource Bottleneck System: SYSA
AUTO(60)
+-------------------------------------+--------------------------------------+
| Resource . . . : CPU | Degraded workload . . . : PROD TSO |
+-------------------------------------+--------------------------------------+

+------------+-------------+-------------------------------------------------+
|   Name     | Perf. Group |    %     ....................................50 |
|------------+-------------+----------+--------------------------------------+
|_ PAYROLL   |  BATCH      |   63.1   |------------------------------------>>|
|_ USERJOB1  |  TSO R&D    |   15.2   |----->                                |
|_ BATJOB1   |  TEST BAT   |    8.3   |--->                                  |
|_ BACKUPS   |  MONITORS   |    7.2   |-->                                   |
|_ USERJOB2  |  TECHSUPT   |    2.2   |->                                    |
|_ BATJOB2   |  TEST BAT   |    2.0   |>                                     |
|_ MONJOB5   |  MONITORS   |    2.0   |>                                     |
|            |             |          |                                      |
+------------+-------------+----------+--------------------------------------+
F1=Help  F2=Keys  F3=Exit  F5=Refresh  F6=Console  F10=Action Bar
F11=Print  F15=System Status
```
Example: Diagnosing a Current Bottleneck

For the purposes of this scenario, let us assume that PAYROLL is not as important right now as improving response time for PROD TSO. You decide to reduce its impact by swapping it out, planning to swap it in later. Five minutes later, you show details for PROD TSO, whose performance has improved dramatically.
Example: Analyzing a Recurring Bottleneck

Introduction

The following is a scenario that shows how you might use OMEGAMON II to analyze a recurring resource contention problem.

Identifying chronic response time problem

Suppose users in the PROD TSO performance group begin to complain that response time has deteriorated over the past few days. They maintain that every afternoon between 3:30 and 4:00, response time is particularly bad. You need to find out why.

Displaying trends for performance group

First, you need to look at a historical trend display for your performance groups, so you request the Show details action for All P.G. OMEGAMON II displays the All Performance Groups panel. Then you request the Historical trends action for PROD TSO, and OMEGAMON II displays the Trend Date/Time Selection pop-up, as shown below. This pop-up enables you to set a date and time range for the current trend panel. This range remains in effect for all subsequent trend panels.
Since you want to limit your trend information to a specific half-hour interval on certain days, enter 4. OMEGAMON II displays the Trends Date/Time Settings pop-up, as shown in the following figure.

**Selecting date/time range**

Enter the time and date range, 3:30 PM to 4:00 PM for the last three days, as shown below. (Instead of entering 03 in the Last Days field, you could also have specified a date range using the Start Date and End Date fields, or by selecting specific days of the week.)

---

**Looking at performance group trends**
Example: Analyzing a Recurring Bottleneck

OMEGAMON II displays the Historical Trends for a TSO Performance Group panel, shown below. As this figure shows, PROD TSO spent most of its time waiting for CPU.

Now you want to see which performance groups were using the CPU when PROD TSO was waiting for it. To do this, open the GoTo pull-down and select **Historical Details for All Performance Groups**.

**Identifying the impactor**

OMEGAMON II displays the Historical Details for All Performance Groups panel, shown below. Now you can identify the impacting performance group. As this figure shows, performance group CICSTEST had the CPU 50% of the time during that interval, while PROD TSO had it only 20% of the time.
Example: Analyzing a Recurring Bottleneck

Taking action
CICSTEST is, as its name implies, a test performance group. Ordinarily, it shouldn’t have a high enough priority to impact PROD TSO. Perhaps someone changed the dispatching priority for CICSTEST.

To check out your hypothesis, you choose Select Start/End Date/Time on the View pull-down and enter another time interval. This shows you all performance groups for that interval. You discover that for each interval, CICSTEST is the main impactor against PROD TSO.

You decide to reduce the impact of CICSTEST by lowering its priority in the IEAIPSxx member of SYS1.PARMLIB. As a result, PROD TSO response time improves dramatically.
Guidelines for Troubleshooting Bottlenecks

Introduction

The execution states reported by OMEGAMON II fall into two categories: *waiting for resources* and *actively using resources*.

The tables in this section describe general symptoms of these two categories of bottlenecks and their possible causes, and suggest actions you might take to alleviate bottleneck situations.

Discovering symptoms

You can discover the bottleneck symptoms described in the following tables by:

- observing the status indicators on the System Status panel and navigating to more detailed panels from there (such as the Details for a Job, Started Task, or TSO User panel or the Analyze Problems or Bottlenecks panels)
- receiving a call from a user reporting an address space (batch job, started task, or TSO user) that isn’t performing as expected and displaying details for that address space

Waiting for resources

The following table describes common situations in which waiting for a resource might become a bottleneck that degrades the performance of a workload. You can recognize these situations by looking at information gathered collectively from the Bottlenecks, Resource Bottlenecks, and Details for a Job, Started Task, or TSO User panels.

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Possible Causes</th>
<th>Possible Solutions</th>
</tr>
</thead>
</table>
| waiting for I/O | waiting for device       | - move datasets to another DASD volume  
|               |                          | - reschedule either job  
<p>|               |                          | - identify the competing workloads and swap them out                                |
| enqueue delay  | enqueue for datasets     | - release enqueue of less important job by canceling the job                        |
| paging waits   | real storage contention  | - storage-isolate the address space                                                |
|               | I/O subsystem is         | - reschedule the job                                                               |
|               | constrained              | - increase minimum system think time                                               |
|               | device contention        | - add a page dataset                                                                |
|               |                          | - ensure there are no other active datasets on page devices                         |</p>
<table>
<thead>
<tr>
<th>Workload Delay</th>
<th>Possible Causes</th>
<th>Recommended Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRM delay</td>
<td>Swaps performed to accommodate other address spaces which are in the same domain as the impacted workload</td>
<td>Make impacted workload nonswappable, select another workload to swap out, raise the MPL of the domain, if storage is available, transfer other workloads out of the domain, as long as the new domain can accommodate them.</td>
</tr>
<tr>
<td>waiting for CPU</td>
<td>Low dispatching priority, looping condition, overcommitted CPU</td>
<td>Adjust dispatching priority, reduce the impact of the other address space by cancelling it, swapping it out, or resetting its performance group to reduce service to it.</td>
</tr>
</tbody>
</table>
### Actively using resources

The following table describes common situations in which actively using a resource might become a bottleneck that degrades the performance of a workload. You can recognize these situations by looking at information gathered collectively from the Bottlenecks, Resource Bottlenecks, and Details for a Job, Started Task, or TSO User panels.

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Possible Causes</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>unusually high or low CPU use</td>
<td>■ looping condition</td>
<td>■ select <strong>Inspect CPU utilization</strong> from the Actions or GoTo pull-downs to see if program is written inefficiently or contains a loop</td>
</tr>
<tr>
<td></td>
<td>■ inefficient coding</td>
<td></td>
</tr>
<tr>
<td>active I/O</td>
<td>■ inefficient placement of datasets on device</td>
<td>■ on non-SMS volumes, move concurrently used datasets closer together</td>
</tr>
<tr>
<td></td>
<td>■ disk fragmentation</td>
<td>■ on SMS-managed volumes, set dataset to must-cache status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ reallocate or compress a dataset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ move dataset to a cache device</td>
</tr>
</tbody>
</table>
Developing Your Own Realtime Screens

Introduction

A screen space is defined as a screen containing a sequence of executable commands, stored as a member under a user-defined name. You type in the command(s) you want to constitute your screen space and OMEGAMON lets you store and invoke the commands in that sequence. Alternatively, you can use a text editor to create screen spaces, since they are treated as members in a partitioned dataset.

This chapter also addresses managing a screen space library, chaining and fetching screen spaces, and automating the process of using screen spaces.

Necessary reference material

The OMEGAMON commands referred to in this manual are explained in the OMEGAMON for MVS Command Language Reference Manual. This section does not attempt to explain commands or keywords presented.

Chapter Contents

Creating/Modifying Your Screen Space .................................................. 158
Steps in Creating or Modifying a Screen Space ........................................ 159
Guidelines for Creating Your Own Screen Spaces .................................. 161
Invoking a Screen Space ........................................................................ 163
Managing Your Screen Space Library .................................................... 164
Listing Screen Spaces ......................................................................... 165
Renaming Screen Spaces ...................................................................... 166
Deleting Screen Spaces ....................................................................... 167
Loading Screen Spaces from Disk to Main Storage ............................... 168
Chaining and Fetching Screen Spaces ................................................... 169
Chaining Screen Spaces Together .......................................................... 170
Conditional Fetching for Screen Spaces ................................................. 171
Manipulating the Log within Screen Spaces ......................................... 172
Delayed Fetching for Screen Spaces ...................................................... 173
Automating and Logging Screen Spaces ................................................. 174
Exception Logging Facility ................................................................... 175
Automatic Screen Facility .................................................................... 176
Timed Screen Facility .......................................................................... 179
Control Log Characteristics for XLF, ASF, and TSF ............................... 180
Creating/Modifying Your Screen Space

You can create a sequence of commands that might be useful to you in your operations, then store it, access it, modify it, and invoke it whenever it is needed. The commands execute the same as if they were entered individually.

In this section

In this section we cover the following topics.

- “Steps in Creating or Modifying a Screen Space” on page 159
- “Guidelines for Creating Your Own Screen Spaces” on page 161
- “Invoking a Screen Space” on page 163
Steps in Creating or Modifying a Screen Space

This section tells how to create your own screen space or modify an existing one. While it is possible to create a screen space with a text editor, here we present the procedure for creating and modifying screen spaces within OMEGAMON II. To create or modify a screen space, follow the steps listed below.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Type <strong>go</strong> (mnemonic for Go to OMEGAMON) in the input field of the action bar. The OMEGAMON Main Menu is displayed. You create a screen space on this panel, the Main Menu panel.</td>
</tr>
<tr>
<td>2</td>
<td>We assume that you are in command mode for these exercises. To go to command mode, press F24 when you are at the OMEGAMON Main Menu. F24 toggles between command mode and menu mode.</td>
</tr>
</tbody>
</table>
| 3    | To create or modify a single screen, use the /DEF ON INFO-line command to place OMEGAMON in definition mode until you are done. Definition mode inhibits updating and prevents OMEGAMON from executing the commands as you enter them.  
Because saving the screen automatically turns off definition mode, use /DEF HOLD if you are creating or modifying several screen spaces in a session. This maintains the session in definition mode until you issue a /DEF OFF command. |
| 4    | To modify an existing screen space, type the name of that screen space on the INFO-line and press Enter. If you do not remember the name of the screen space, use the SCRN immediate command to see a listing of screen spaces. |
| 5    | To create a new screen space, clear the screen to make space to enter commands by using the clear screen immediate command. Enter ..**bb** in the input area of any line below the INFO-line. This clears all of the logical screen below the command. |
| 6    | Enter the desired OMEGAMON commands on the screen. Refer to “Guidelines for Creating Your Own Screen Spaces” on page 161 later in this section for additional information on formatting and managing the screen space. |
| 7    | - If you have created a new screen, either from scratch or by modifying an old one, type **SAVE cccccccc** on the INFO-line, where **cccccccc** is the name you want to give your new screen, and press Enter.  
- If you have modified an existing screen and want to write those modifications to the same screen, type **/REP cccccccc** (Replace command) instead of **/SAVE** in order to save the modifications.  
The new contents are saved in the same screen space name.  
OMEGAMON saves the screen as the member identified with the RKOMPCSV ddname as specified in the OMEGAMON start-up CLIST or the OMEGAMON started task PROC. |
Steps in Creating or Modifying a Screen Space

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Use /DEF OFF to restore automatic updating, if you turned on definition mode with the HOLD option.</td>
</tr>
<tr>
<td>9</td>
<td>To return to OMEGAMON II, type /STOP on the INFO line.</td>
</tr>
</tbody>
</table>

**Example of a screen space**

The following example shows a screen space named SAMPLE. Notice the command explanations in the divider lines and the additional comments at the bottom.

```
SAMPLE VTAM LOG OM/DEX. V750./C A083. mm/dd/yy 08:31:41
>*************************************************** SAMPLE - System Paging Rates ***************************************************
SYS
>******************************************************** Paging: Total system ------------------------------------------>
> $SPAL.R
>******************************************************** Paging: System area (private and common) ------------------------------------------>
> $SPS .R
>******************************************************** Paging: Due to swapping ------------------------------------------>
> $SPWS.R
>******************************************************** Paging: Total common area ------------------------------------------>
> $SPC .R
>******************************************************** Paging: Total common area page-ins ------------------------------------------>
$SPCI.R
>******************************************************** To change the scale of the plot, use the GSCL immediate command. ==
>NOTES: To change the scale of the plot, use the GSCL immediate command.
```
Guidelines for Creating Your Own Screen Spaces

Guidelines to follow
The guidelines below are designed to help you avoid some common pitfalls.

<table>
<thead>
<tr>
<th>Guideline for</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition mode</td>
<td>Use definition mode when creating or modifying your screen space so that commands will not execute as you enter them.</td>
</tr>
<tr>
<td>INFO-line commands</td>
<td>You cannot include INFO-line commands as part of the commands on the screen. Instead, use the equivalent immediate command.</td>
</tr>
<tr>
<td>Cursor position</td>
<td>The default cursor position is the first position of the INFO-line. If you want to save the cursor in a particular position on the screen, type /REP or /SAVE on the INFO-line, move the cursor to the desired position, and then press Enter to save the screen as formatted. The cursor will display in that position whenever OMEGAMON invokes the screen. If you use a text editor to modify screen spaces and want to change the cursor position, use the CURS=(n,m) command.</td>
</tr>
<tr>
<td>Scrolling</td>
<td>Use F7 (or F19) for scrolling up and F8 (or F20) for scrolling down one physical screen at a time. Change the default scroll amount with the OPTN command. Or use /TOP or /BOTTOM INFO-line commands to scroll to the top or bottom of a logical screen.</td>
</tr>
<tr>
<td>Separator lines</td>
<td>Include separator lines to make the screen more readable. OMEGAMON automatically draws a separator line across the screen with the c====aa immediate command. The variable c allows you to specify the color if you have extended color on your terminal, and the variable aa allows you to specify the characters you want to use for the line.</td>
</tr>
<tr>
<td>Comment lines</td>
<td>When you insert comment lines, begin all lines of comment text with a greater than sign (&gt;) in column 1. A command included in a comment line will not execute.</td>
</tr>
<tr>
<td>Lines that comment themselves out</td>
<td>Some OMEGAMON commands are designed to comment themselves out after they execute. Make sure that the comment character (&gt;) does not appear in column 1 when you save the screen. Using definition mode when creating the screen space alleviates this problem.</td>
</tr>
</tbody>
</table>
### Guidelines for Creating Your Own Screen Spaces

**Inserting blank lines**  
To insert blank lines into a screen, use the `.I nn` immediate command. The variable `nn` indicates the number of lines to insert. The default is 1. Note that the `nn` argument must be in columns 4-7.  
OMEGAMON inserts the new lines above the line where you type the insert command. Therefore, all other lines currently on the screen below the inserted line shift downward. When the command executes, the line you typed over with the insert command restores to its original data.

**Deleting lines**  
To delete lines from a screen space, use the `.D nn` immediate command, which deletes the number of lines specified in `nn` from the current line. When you do not specify `nn`, it defaults to 1 deletion line.  
You can also delete a block of data from the logical screen. To do this, enter `.DDb` on the first and last lines of the block.

**Assigning a function key to a screen space**  
If you have created a particularly useful screen space, you can assign a function key to that screen space for the current OMEGAMON session with the `.PFK` command. For example, type `.PFK09=DISKS` which assigns function key 09 to the screen space named DISKS for the duration of the current session.  
To assign a function key to a screen space permanently, refer to the *OMEGAMON for MVS Command Language Reference Manual.*
Invoking a Screen Space

When you want to display the output generated by a screen space, just invoke the screen space.

**Invoking by name**

To invoke a screen space, type the screen name on the INFO-line and press Enter.

**Invoking by using a function key**

In your current session you can assign a screen space to a function key and invoke it by pressing that function key. The function key will remain assigned to the key only for the duration of your session.

**What happens when you invoke a screen**

When you invoke a screen with your commands, it executes the commands in the screen space and replaces the contents of the current screen with their output.
Managing Your Screen Space Library

Introduction
You now know how to create a screen space and save it in a library. To manage your

In this section
In this section we cover the following topics.

- “Listing Screen Spaces” on page 165
- “Renaming Screen Spaces” on page 166
- “Deleting Screen Spaces” on page 167
- “Loading Screen Spaces from Disk to Main Storage” on page 168
Listing Screen Spaces

The SCRN immediate command, used without an argument, lists all screen spaces in main storage and in the disk datasets pointed to by the RKOMPCSV and RKOMPROC ddnames.

You can list screen spaces by using arguments to request a list of screen spaces in main storage, disk, or both. You can also specify alphabetic limits for a range of member names, and wildcards are allowed.

Example of a screen space listing

The next figure shows a typical Candle-supplied screen space listing, produced with the SCRN immediate command. OMEGAMON sorts the list by dataset, with in-storage screen spaces first, followed by RKOMPCSV, then each dataset in the RKOMPROC concatenation.

```
>SCRNB $ thru 99999999
> In-storage screen facility 20 members 4232 bytes used
> $ #01 #02 #03 DEBUG DEBUG01 DEBUG02 DEBUG03
...
> RKOMPCSV rhilev.RKOMPCSV
> #01 SAMPLE VSAM
> RKOMPROC (DUP.) rhilev.RKOMPCSV
> RKOMPROC +01 rhilev.RKOMPROC
> #01 #07 #08 #09 #10 #13 #14 #15
> #16 #17 #18 #19 #20 #21 #22 #23
> #24 BITMAP CPAM CSA DD DDACVOPS DDBASIC DDEX
> DDEXCEPT DDIO DDNAMES DDNEWS DDOMEGA DDOPER DDPFK DDSYS
> DDSYE DDSYSI DDTASK DEBUG DEBUG01 DEBUG02 DEBUG03 DEBUG04
> DEXALL DEXAL2 DEXAL3 DEXAL4 DEXAL5 DEXAN DEXDFT DEXDF2
> DSA ENQUEUES EOD HELP HELP2 INTERVAL M2INIT00 PAM
> SCREEN SCREENS SIT SLOT SLOTS STOP STOPI STOPC STOPP
> SUBPOOLS TABLES TASK TASKS VIRTUAL VSAM
```
Renaming Screen Spaces

Use the RENM immediate command to rename a screen space.

Procedure for renaming

The RENM command accepts an argument specifying where the screen space is to be found. To rename the SAMPLE screen space to EXAMPLE in both main storage and RKOMPCSV, enter:

```
RENM SAMPLE EXAMPLE
```

The following message appears:

```
> >> Member "SAMPLE " Renamed to "EXAMPLE " Both In-Storage and in RKOMPCSV <<
```
Deleting Screen Spaces

Use the DELT immediate command to delete screen spaces.

Rules for deleting
OMEGAMON deletes screen spaces only from main storage and the dataset pointed to by the RKOMPCSV ddname, not from the dataset pointed to by the RKOMPROC ddname, the read-only dataset. The DELT command accepts an argument specifying where to find the screen space to be deleted.

Procedure for deleting
To delete the screen space SAMPLE from both main storage and RKOMPCSV, enter:

   DELT SAMPLE

The following message appears:

>   >> Member "SAMPLE " Deleted Both In-Storage and from RKOMPCSV <<
Loading Screen Spaces from Disk to Main Storage

The LSCR immediate command loads screen spaces from disk to main storage. By placing screen spaces in storage with LSCR, execution is quicker. This capability is especially useful for dedicated mode, since if an I/O subsystem is unavailable, your screen space will not be available.

Procedure for loading

The format for loading your screen space is:

```
LSCR  cccccccc  cccccccc  ...  cccccccc
```

The variables are screen space names. You can load as many screen spaces as can fit on the input line. Continuation to the next line is allowed.

In the following example, OMEGAMON loads screen spaces **ZZ1**, **ZZ2**, and **ZZ3** from disk to main storage.

```
LSCR  ZZ1 ZZ2 ZZ3
```
Chaining and Fetching Screen Spaces

Introduction

You can chain screen spaces together so that a series of screens can be invoked automatically. You can also fetch specific screens based upon user-defined conditions or time delays.

In this section we cover the following topics.

- “Chaining Screen Spaces Together” on page 170
- “Conditional Fetching for Screen Spaces” on page 171
- “Manipulating the Log within Screen Spaces” on page 172
- “Delayed Fetching for Screen Spaces” on page 173
Chaining Screen Spaces Together

You can chain a group of screen spaces together, so that an entire series of screen spaces can execute without operator intervention.

Using .FGO and .SGO

A .SGO or .FGO immediate command, entered in a pre-defined screen space, fetches the next screen automatically. The .FGO (fast go) and .SGO (screen go) commands perform similar functions. .SGO fetches the specified screen space on the next screen refresh and displays the screen spaces as they execute. .FGO allows fast access to screen spaces within the current screen refresh and bypasses the display.

Beware of looping screen spaces

An improper sequence of .FGO screen spaces could cause a looping condition. To protect against looping, OMEGAMON limits the number of consecutive fetches allowed (64 by default). After reaching the limit, .FGO acts like .SGO so that executing screen spaces display on each screen refresh. Therefore, if .FGO screen spaces cause a loop, you need to correct the condition and re-enable .FGO.

To re-enable .FGO, reset the FGOLOOP keyword of the .SET command. The FGOLOOP is set to OFF until the limit is reached. Then OMEGAMON turns it ON, indicating the probability of a loop.

Testing .FGO routines

FGOLOOP=OFF causes .FGO to bypass the terminal display. When testing, you may want to see what is happening. Set FGOLOOP=ON with the .SET command to test your screen space chaining routines.
Conditional Fetching for Screen Spaces

The .FGO and .SGO commands give you the flexibility of conditional screen space fetching. You can fetch a screen space if and only if certain conditions are true. Arguments, which may include relational operators, can be used. You can compare the CPU serial number, the mode of operation, the operating system level, the profile in use, the SMF ID, the OMEGAMON console address, or any variable you have set with the .VAR command.

Example of conditional fetching

For example, to fetch screen space SAMPLE if and only if you are running in an XA environment, enter:

.SGO SAMPLE OPSYS=210
Manipulating the Log within Screen Spaces

Conditional screen space fetching enables you to control the state of the log while branching.

Logging screen spaces selectively

To enable you to log screen spaces selectively, the .LOG immediate command has PUSH and POP arguments. You can include these arguments in screen spaces to change the on or off status of the log when you branch to other screen spaces, and then return the log to its original state after the screen routines are completed.

Example of .LOG

For example, if you had a series of four screen spaces chained together with .FGO and you wanted to log only the last two, you could use .LOGPUSH and .LOGPOP in the following manner:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>At the end of screen space 2, enter <strong>.LOGPUSH</strong> to have OMEGAMON record the previous status of the log. Then, enter <strong>.LOGON</strong> to turn on the log at the beginning of screen space 3.</td>
</tr>
<tr>
<td>2</td>
<td>Enter <strong>.LOGPOP</strong> at the end of screen space 4. Thus, the log will be restored to its original status when your branching routine is complete.</td>
</tr>
</tbody>
</table>

The .LOGPUSH and .LOGPOP functions are performed automatically when you log screen spaces using the Exception Logging Facility (XLF), the Automatic Screen Facility (ASF), or the Timed Screen Facility (TSF), discussed later in this chapter.
Delayed Fetching for Screen Spaces

Both n.FGO or n.SGO let you delay the fetching of a screen space for a specified number of OMEGAMON screen refreshes.

Why delay fetching a screen space
Delayed fetching allows you to accommodate commands that take more to execute than the time allocated for one screen refresh interval. It also permits you to log several successive executions of a given screen space before moving on to the next screen space. The n in front of .FGO or .SGO specifies the number of screen refreshes that constitute the delay.

Format of delayed fetching
The format for delayed fetching is:

```
.FGO screenname
.SGO screenname
```

The variable n can be a number representing a delay of 1 to 9 screen refreshes or a letter from A to Z representing a delay of 10 to 35 screen refreshes.
Automating and Logging Screen Spaces

Through Exception Logging Facility (XLF), Automatic Screen Facility (ASF), and Timed Screen Facility (TSF) in OMEGAMON, you can automatically invoke a screen space, initiate action, and log information in response to exception conditions or user-specified time.

How do XLF, ASF, and TSF work?
The power of these automating and logging facilities lies in the ability to branch to, execute, and log a series of screens. The screen spaces invoked can contain whatever information-gathering and/or action-taking commands you specify. You might include commands to turn on the log, change OMEGAMON defaults, further analyze an exception condition, or call other screen spaces. When the sequence is finished, OMEGAMON then resumes normal operation -- all without manual intervention.

What are the differences between them?
XLF, ASF, and TSF all invoke screen spaces. Both XLF and ASF are based on the occurrence of an exception condition. Only XLF prints out an exception message. TSF invokes screen spaces based on user-specified time only.

Assumption
This section assumes you are familiar with exception analysis and how to create screen spaces.

In this section
In this section we cover the following topics.

- “Exception Logging Facility” on page 175
- “Automatic Screen Facility” on page 176
- “Timed Screen Facility” on page 179
- “Control Log Characteristics for XLF, ASF, and TSF” on page 180
Exception Logging Facility

The Exception Logging Facility (XLF) automatically time-stamps and logs exception messages for your review. It enables you to correct intermittent performance problems by documenting the frequency and severity of exceptions.

Commands required
The commands that you need to set up and operate XLF are XACB, OPTN, and /XLF.

Setting parameters for XLF operation
The parameters for XLF operation are set with the XACB command. XACB allows you to activate the XLF features for all occurrences of any given exception, or only if the exception persists for a specified number of screen refreshes. You can also set a limit on the number of times a given exception invokes the XLF feature. Refer to the OMEGAMON for MVS Command Language Reference Manual for a description of the XACB command and parameters.

How to use XLF
The procedure for using XLF is given in the following table.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create or find the member name of any screen spaces you want to invoke. Use the .SGO or .FGO command to chain screens together. Use the .RTN command on the last screen to return control to OMEGAMON.</td>
</tr>
<tr>
<td>2</td>
<td>Use the XSUM command to display a summary of exceptions and current status.</td>
</tr>
<tr>
<td>3</td>
<td>Use the XACB command to update an entry.</td>
</tr>
<tr>
<td>4</td>
<td>Use the OPTN command to turn on the facility.</td>
</tr>
</tbody>
</table>
Automatic Screen Facility

The Automatic Screen Facility (ASF) automatically invokes a predefined screen space when a given exception occurs for more than a specified number of successive screen refreshes. The predefined screen space can contain commands to turn on the log, further evaluate the exception condition, and perform other options.

ASF functions only in dedicated mode or VTAM mode with automatic update in effect.

Required commands
The commands that you need to set up and operate ASF are XACB, OPTN, and /LOG.

Setting parameters for ASF
The parameters for ASF operation are set with the XACB command. XACB allows you to activate the ASF features for all occurrences of any given exception, or only if the exception persists for a specified number of screen refreshes. You can also set a limit on the number of times a given exception invokes the ASF feature. Refer to the OMEGAMON for MVS Command Language Reference Manual for a description of the XACB command and parameters.

How to use ASF
The procedure for using ASF is given in the following table.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create or find the member name of any screen spaces you want to invoke. Use the .SGO or .FGO command to chain screens together. Use the .RTN command on the last screen to return control to OMEGAMON.</td>
</tr>
<tr>
<td>2</td>
<td>Use the XSUM command to display a summary of exceptions and current status.</td>
</tr>
<tr>
<td>3</td>
<td>Use the XACB command to update an entry</td>
</tr>
<tr>
<td>4</td>
<td>Use the OPTN command to turn on the facility.</td>
</tr>
</tbody>
</table>

Example of ASF
In this example we assume that a screen space has already been created and you go to it using the XACB command. You must set the ASF keyword of the OPTN command to ON to enable the automatic screen feature.
Here are some parameters set with the XACB command for the DRDY exception.

- XLF Parameters:
  - Auto=ON
  - Log=OFF
  - Limit=3
  - Repeat=NO
  - Persist=5
  - SL=DEX01

In this example, when the exception condition exists for a duration of five refreshes, ASF invokes screen space DEX01. From this point on, an A appears in the far right portion of the INFO-line, which indicates that the current screen is part of an ASF sequence.

DEX01 might consist of the following sequence of commands:

```
DEX01 DED LOG OM/DEX. V750./C A083 mm/dd/yy 17 :03:37 A

> After 8 screen refreshes, branch to screen space DEX02
8.SGO DEX02

> Reset the OMEGAMON screen refresh time to 15 seconds
.SET INTERVAL=15

> Start degradation analysis
DEX
BCGN

> Display degradation analysis for performance group 2
PCT02
```

In this example, the OMEGAMON screen refresh time is set to 15 seconds. Since .SGO waits eight screen refreshes before it jumps to DEX02, bottleneck analysis runs on this screen for 2 minutes and (following the SL= parameter) logs the results. At this point, screen space DEX02, shown below, is invoked.
This screen space suspends degradation analysis, and returns to the calling screen with the .RTN command. Upon return, the ASF sequence terminates (the A disappears from the INFO-line). .RTN also automatically resets the interval to the one in effect when the ASF sequence started. (You could, however, add an NR argument to .RTN (.RTNNR) to direct OMEGAMON to keep the new interval in effect.)

**Note:** Because the \texttt{SL=} parameter was used, ASF automatically turned on the REPORT log when the exception occurred (if it was not already on). When you leave automatic mode, ASF also turns off the log (if it was in the off state before the exception tripped).

When you use ASF to turn on the log automatically with \texttt{SL=}, OMEGAMON first logs the screen in use, and then branches to the scheduled screen space. This is done so that any exceptions can be logged before the ASF sequence begins.
Timed Screen Facility

The Timed Screen Facility (TSF) automatically invokes screen spaces at specified times (for example, at 2200 hours) or at specified time intervals (for example, every 15 minutes). Many sites use TSF to spin off copies of the REPORT and/or of XLFLOG files to the printer. You can use the TSF facility to automate many day-to-day housekeeping routines.

Screen auto refresh

*Note:* If you want TSF to be operational, do not turn off screen automatic refresh.

Required Commands
The commands that you need to set and operate TSF are .TSF, OPTN, and /LOG.

How to use TSF

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create or find the member name of any screen spaces you want to invoke. Use the .SGO or .FGO command to chain screens together. Use the .RTN command on the last screen to return control to OMEGAMON.</td>
</tr>
<tr>
<td>2</td>
<td>Enter the .TSF00 command to list current entries of the TSF feature (ON/OFF).</td>
</tr>
<tr>
<td>3</td>
<td>To update an entry, type <code>.TSFnn</code> over the current entry in the TSF table, and press Enter.</td>
</tr>
<tr>
<td>4</td>
<td>Use the OPTN command to turn on the facility.</td>
</tr>
</tbody>
</table>
Control Log Characteristics for XLF, ASF, and TSF

You can display or change the logging parameters for XLF, ASF, and TSF with the OUTP major command and its minors. Refer to the OMEGAMON for MVS Command Language Reference Manual for a description of the minor commands for OUTP.

OUTP command parameters

- For XLF, specify OUTP XLFLOG.
- For ASF or TSF, specify OUTP REPORT.

When does logging appear?

In XLF and ASF processing, only one record is written to the log while a given exception condition persists. However, new records will be written to the XLFLOG or the REPORT log if any of the following situations occur:

- If an exception disappears for even one screen refresh, and then reappears, it is considered a new event.
- If you turn a given exception off and then on again, you clear the event. If the exception condition still exists, then ASF or XLF will be invoked again. OMEGAMON does not check for the condition unless the exception is turned on.
- If you turn off XLF or ASF and then turn it on again, you clear all events and new records are written to the log.
- For a given exception, if you set the XACB REPEAT parameter to YES and specify a threshold for the PERSIST parameter, a new record is written each time the PERSIST threshold is reached.
Gaining Historical Perspective

Introduction

OMEGAMON II realtime panels help you monitor the health of your system as events occur. OMEGAMON II also provides historical information about your system’s workloads and resources. By studying the information gathered from a historical perspective, you can:

- determine whether current performance is standard or problematic
- identify trends

You can display IPS-based historical information using OMEGAMON II action codes, pull-down menus, batch reporter panels, and the Workload Profiling Facility. In this chapter we describe the procedures necessary to access this historical information, both online and in printed form.

MVS/SP 5 and historical data

When navigating to a historical panel, you sometimes receive a message indicating that no records were found for the time period you requested. Under MVS/SP 5, this can mean that MVS was running in goal mode during the requested period and only WLM-based data is available.

Chapter Contents

- Accessing Batch Report Controls .................................................. 192
- Identifying Critical Workloads for Profiling ................................... 199
Displaying Historical Information Online

You can display OMEGAMON II historical detail and trend information online by using action codes and pull-down menus from any OMEGAMON II panel. In this section, we describe how to use these action codes and pull-down menus, and how to set various controls to select the most appropriate detail and trend information.

What are historical details?

Historical detail panels provide information for a specified time period. For instance, if you notice a current DASD problem and you would like to see how long it has been occurring, you might display a historical DASD panel showing details for the last three hours. The resulting panel combines historical DASD information into a single averaged result.

What are historical trends?

In contrast to historical detail panels, historical trend panels portray trends over a series of time intervals. For instance, to expand on the DASD historical details example described above, you might display a historical DASD trend panel to look at DASD activity during the same time period, but for every day for the past week.

Setting Controls for Online Historical Panels

The following controls help you select the most appropriate information for your historical detail and trend panels:

- history refresh interval
- date and time range for trend panels
- date and time range for detail panels

These controls are described below.

Setting history refresh interval

Historical data is typically collected to a historical datastore at the end of every RMF interval. This data is not available for historical reporting in OMEGAMON II until a refresh is performed. A refresh is the frequency with which OMEGAMON II checks the historical datastore for newly collected data.

The default refresh interval is 60 minutes. If the RMF interval on your system is set to 15 minutes, this means that a refresh will occur every fourth RMF interval, which ensures that no more than four RMF intervals will elapse before historical data is available to OMEGAMON II.
The value you choose as a refresh interval depends on how current you want the data in your historical reports to be. Although a refresh interval shorter than 60 minutes requires more overhead than one which is longer, it ensures more current data. It’s up to you.

- To set the history refresh interval for historical panels, select Historical Online/Batch Controls from the Options pull-down, then select **History Refresh Interval Selection** (or enter fast path **ohr**).

**Setting detail date and time**
- To set a date and time range for historical detail panels, select Historical Online/Batch Controls from the Options pull-down, then select **History date/time selections** (or enter fast path **ohh**).

**Setting trend date and time**
- To set a date and time range for historical trend panels, select Historical Online/Batch Controls from the Options pull-down, then select **Trend date/time selections** (or enter fast path **oht**).

**Displaying Historical Detail Information**
You can display panels containing historical detail information by using action codes and pull-down menus.

**Access methods**
There are three ways to display historical detail panels.

<table>
<thead>
<tr>
<th>Access Method</th>
<th>Is Available From</th>
<th>And Results In</th>
<th>Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical detail</td>
<td>any workload or resource light on the System Status panel,</td>
<td>historical details about a workload or resource for a specified time</td>
<td>h</td>
</tr>
<tr>
<td>action code</td>
<td>or on a detail line of subsequent panels</td>
<td>period.</td>
<td></td>
</tr>
<tr>
<td>GoTo pull-down</td>
<td>action bar</td>
<td>selections for related historical panels.</td>
<td>g</td>
</tr>
<tr>
<td>Index pull-down</td>
<td>action bar</td>
<td>a list of historical detail categories.</td>
<td>ih</td>
</tr>
</tbody>
</table>

We will discuss each access method in turn.
Example of H action code

One way to display a historical detail panel is to use the H action code.

Suppose you are monitoring your system from the System Status panel when you notice that the STC/APPC status indicator is red. You request the **Show details** action for this indicator to display the Started Tasks Overview panel, as shown in the following figure.

When you enter `vp` in the action bar input field to view problem started tasks, you see that there are currently eight started tasks that have high CPU usage or that are waiting to execute. You wonder if started task ABC0011, whose current CPU usage is 9.7%, has been using that much CPU time for the last few hours.

Using fast path `ohh`, you first provide a date and time range of the last few hours of today. Then, you enter `h` next to the started task ABC0011 to request historical details for that task, as shown in the example above. The Historical Details for Batch Job, STC or TSO User panel displays, as shown in the following figure.

According to the information on this panel, job ABC0011 spent 78% of its time during the last three hours using the CPU. This may have caused a bottleneck for other tasks trying to gain CPU resources. Depending upon the importance of this started task, you may decide to constrain its CPU usage by changing its performance group, for instance.
Example of GoTo pull-down

Another way to display a historical detail panel is to use the GoTo pull-down. In this example, selection 8 displays historical detail information for overall system performance.
Example of Index pull-down

A third way to display a historical detail panel is to use the Index pull-down. The following is an example of an Index pull-down selection that contains categories for historical detail panels. In this example, displayed by entering fast path `ih`, three categories of historical detail panels are available -- workloads, I/O, and system. After you select a category, OMEGAMON II displays a list of the historical detail panels for that category.
Displaying Historical Trend Information

You can display panels containing historical trend information by using action codes and pull-down menus.

Access methods

There are three ways to display historical trend panels.

<table>
<thead>
<tr>
<th>Access Method</th>
<th>Is Available From</th>
<th>And Results In</th>
<th>Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical trend action code</td>
<td>any workload or resource light on the System Status panel, or on a detail line of subsequent panels</td>
<td>historical trends about a workload or resource over a series of time intervals.</td>
<td>t</td>
</tr>
<tr>
<td>GoTo pull-down</td>
<td>action bar</td>
<td>selections for related historical panels.</td>
<td>g</td>
</tr>
<tr>
<td>Index pull-down</td>
<td>action bar</td>
<td>a list of historical trend categories.</td>
<td>it</td>
</tr>
</tbody>
</table>

We will discuss each access method in turn.

Example of T action code
One way to display a historical trend panel is to use the T action code.

Suppose you receive a complaint from your TSO users that response time is poor between 3:30 and 4:00 PM every day. You need to find out why.

First, you request a historical trend display for your TSO performance groups by entering t next to the TSO Host status light on the System Status panel. Then, you provide the number of a TSO performance group and a date and time range of 3:30 to 4:00 PM every day for the last week.

OMEGAMON II displays the Historical Trends for a TSO Performance Group panel, as shown in the following figure.

As this panel shows, the main bottleneck is private page-in. This is just as you suspected, since you know that paging delays are frequently at the root of poor TSO response time. However, you decide to investigate a little further to see if your hypothesis is supported.

You decide to view details for real storage use and paging devices, and check your System Resources Manager (SRM) values. To do this, select the following panels from the Index pull-down path.

<table>
<thead>
<tr>
<th>Panel</th>
<th>Fast Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Storage Utilization</td>
<td>ispr</td>
</tr>
<tr>
<td>Paging Activity</td>
<td>isap</td>
</tr>
<tr>
<td>SRM Information</td>
<td>ikss</td>
</tr>
</tbody>
</table>

If all of these areas point to contention for real storage (and, therefore, to paging delays), there are several long and short term hardware and software solutions. Long-term hardware
Short-term software solutions include:

- checking the IEAIPSxx member of SYS1.PARMLIB to see if storage isolation has been specified for first period transactions.
- setting entries in the IEAOPTxx member of SYS1.PARMLIB to allow SRM to adjust the multiprogramming level (MPL) when paging is high.

**Example of GoTo pull-down**

Another way to display a historical trend panel is to use the GoTo pull-down. The following is an example of a GoTo pull-down containing selections for historical panels. In this example, selection 7 displays historical trend information for system performance.

<table>
<thead>
<tr>
<th>g__ Actions</th>
<th>GoTo</th>
<th>Index</th>
<th>Options</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM2001D</td>
<td>_</td>
<td>1. CPU Utilization</td>
<td>YSA</td>
<td>YSA</td>
</tr>
<tr>
<td>_ Workload</td>
<td>_</td>
<td>2. Details for a Batch Job or Started Task</td>
<td>60)</td>
<td>60)</td>
</tr>
<tr>
<td>_ Batch</td>
<td>_</td>
<td>3. Enqueue and Reserve Conflicts</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>_</td>
<td>4. Paging Activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_ STC/APP</td>
<td>_</td>
<td>5. Common Storage Utilization</td>
<td>---+</td>
<td>---+</td>
</tr>
<tr>
<td>_ TSO: RT</td>
<td>_</td>
<td>6. DASD Response and Percent Busy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_ All P.G.</td>
<td>_</td>
<td>7. Historical Trends for System Performance</td>
<td>---+</td>
<td>---+</td>
</tr>
<tr>
<td>_ Domains</td>
<td>_</td>
<td>8. Historical Details for Overall System Performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>_</td>
<td>9. Enter Omegamon Commands</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>_</td>
<td>10. Enter Epilog Commands</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>_</td>
<td>F1=Help F12=Cancel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example of Index pull-down**

A third way to display a historical trend panel is to use the Index pull-down. The following is an example of an Index pull-down selection that contains categories for historical trend panels. In this example, displayed by entering fast path **it**, three categories of historical trend panels are available -- workloads, I/O, and system. After you select a category, OMEGAMON II displays a list of the historical trend panels for that category.
Generating Printed Historical Reports

Using OMEGAMON II for MVS batch reporter panels, you can submit a batch job to generate printed reports about MVS workloads and system resources.

Printed vs. online reports

With online historical panels, you see a screenful of information at a time, requiring you to scroll from one screen to the next. Printed reports enable you to see entire pages of information at a glance, and allow you to copy and distribute a permanent record of the information. This section describes how to obtain printed reports.

Types of workload reports

The following types of workload reports are available.

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload Summary</td>
<td>Summarizes the condition of a workload on a bar graph. Includes the workload’s average response and/or the availability of data during each one-hour interval.</td>
</tr>
</tbody>
</table>
The following types of resource reports are available.

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload Detail</td>
<td>Provides detail about a workload’s bottlenecks, in tabular form.</td>
</tr>
<tr>
<td>Workload Exception</td>
<td>Provides information about workloads whose service levels are not being met, in tabular form.</td>
</tr>
<tr>
<td>Resource Report</td>
<td>Shows the utilization of a system resource, on a bar graph.</td>
</tr>
<tr>
<td>Resource Detail</td>
<td>Provides detailed information about a system resource, in tabular form.</td>
</tr>
<tr>
<td>Resource Exception</td>
<td>Provides information about a system resource whose expected service levels are not being met, in tabular form.</td>
</tr>
</tbody>
</table>
Accessing Batch Report Controls

By setting batch report controls, you specify the types of reports you want to generate, define the format of various elements of the report, and determine when to submit the batch generation job.

The following paragraphs describe how to access the OMEGAMON II panel that enables you to set these controls.

Accessing reporting options: fast path
Fast path ohb accesses the Batch Reports Controls pop-up. Select a recipient to access the Individual Batch Reporting Options pop-up, where you set the batch report controls.

Accessing reporting options: detailed steps
Following are detailed steps describing how to access the Individual Batch Reporting Options pop-up.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select <strong>Historical online/batch controls</strong> from the Options pull-down menu.</td>
</tr>
<tr>
<td>2</td>
<td>Select <strong>Batch reports controls</strong>. OMEGAMON II for MVS displays the Batch Reports Controls pop-up window.</td>
</tr>
</tbody>
</table>
| 3    | Perform the appropriate action:  
|      | ■ If your name appears on the pop-up as a recipient, type **s** next to the name and press Enter.  
|      | ■ If not, type your name on the top line and press Enter. |

The Individual Batch Reporting Options pop-up is displayed, with your name entered as the recipient.

Editing Batch Report Controls
The following paragraphs describe how to edit these reporting options on the Individual Batch Reporting Options pop-up:

- date format
- report page length
- JCL and print parameters
- file names

Selecting a date format
Select a date format by entering **1** or **2** in the Date Format field. Option 1 (MM/DD/YY) is the default.
Specifying report page length
Specify the report page length by entering the number of lines in the Page Length field. The default is 60 lines.

Specifying JCL and print parameters
Perform the following steps to create a job statement and set print options.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select <strong>JCL and Print Parameters</strong>. OMEGAMON II for MVS displays the JCL Information pop-up.</td>
</tr>
<tr>
<td>2</td>
<td>Fill in your job statement information.</td>
</tr>
<tr>
<td>3</td>
<td>Select a password option to satisfy your site’s requirements.</td>
</tr>
<tr>
<td>4</td>
<td>Fill in the print parameters, and press Enter. OMEGAMON II for MVS returns to the Individual Batch Reporting Options pop-up.</td>
</tr>
</tbody>
</table>

Specifying file names
The default DD statement file names for your site are in the Batch Reporter procedure included in your site’s procedure library. The default name of this procedure is OMIIBATR. See your system administrator for more information.

To edit the default values, perform the following steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select <strong>File Names</strong>. OMEGAMON II for MVS displays the DD Statement File Name Options pop-up.</td>
</tr>
<tr>
<td>2</td>
<td>Type the new dataset names in the spaces provided as necessary, and press Enter. OMEGAMON II for MVS returns to the Individual Batch Reporting Options pop-up.</td>
</tr>
</tbody>
</table>

Selecting Reports
After setting batch report controls such as date format and print parameters on the Individual Batch Reporting Options pop-up, you specify the types of reports (workload or resource) you want to generate. The following paragraphs describe how to select these reports.
Selecting workload reports
To select workload reports, follow these steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select the workload report(s) you want to generate and press Enter. OMEGAMON II for MVS displays the Workload Report Options pop-up for the type of workload report you selected.</td>
</tr>
<tr>
<td>2</td>
<td>Enter the four-character SMF system ID of the system about which you want to report.</td>
</tr>
<tr>
<td>3</td>
<td>Select <strong>Date Control</strong>, and press Enter. OMEGAMON II for MVS displays the Batch Reporting Date Options pop-up for workload summary reports, and the Batch Reporting Date/Time Options pop-up for workload detail and exception reports.</td>
</tr>
<tr>
<td>4</td>
<td>Choose the timeframe you want your workload report to cover, and press Enter. OMEGAMON II for MVS returns to the options pop-up for the type of workload report you selected.</td>
</tr>
<tr>
<td>5</td>
<td>Enter the name of a workload on the Workload Report Options pop-up.</td>
</tr>
<tr>
<td>6</td>
<td>For detail and exception reports, if the workload is a TSO performance group, specify which period to include.</td>
</tr>
<tr>
<td>7</td>
<td>Return to the Individual Batch Reporting Options pop-up by pressing F12.</td>
</tr>
</tbody>
</table>

OMEGAMON II for MVS processes your changes and sets the status for each selected report type to **Included**.

Selecting resource reports
To select resource reports, follow these steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select the resource report(s) you want to generate and press Enter. OMEGAMON II for MVS displays the Resource Report Options pop-up for the type of resource report you selected.</td>
</tr>
<tr>
<td>2</td>
<td>Enter the four-character SMF system ID of the system about which you want to report.</td>
</tr>
<tr>
<td>3</td>
<td>Select <strong>Date Control</strong>, and press Enter. OMEGAMON II for MVS displays the Batch Reporting Date Options pop-up for the Resource Report, and the Batch Reporting Date/Time Options pop-up for resource detail and exception reports.</td>
</tr>
</tbody>
</table>
OMEGAMON II for MVS processes your changes and sets the status for each selected report type to Included.

### Scheduling the Batch Job

All controls you specified in “Editing Batch Report Controls” on page 192 and “Selecting Reports” on page 193 can be saved in a dataset which you can later submit as input to a batch job scheduling procedure. This optional step enables you to defer the generation of your printed reports. You can resubmit this job as many times as you want, without having to reset the batch report controls.

**Before you begin**

Before you begin, be sure that you have already allocated the dataset you are using, and that the OMEGAMON II for MVS started task has write access to it. See your system administrator if you need help.

**Procedure**

On the Individual Batch Reporting Options pop-up, save the JCL information by entering a fully qualified, partitioned dataset and member name.

Once you have saved the jobstream, you can browse and edit the JCL using ISPF.

### Generating Reports

In “Scheduling the Batch Job” on page 195, you learned how to save the JCL so you can defer the generation of the printed reports. The following paragraphs describe how to specify when to submit the batch generation job.

**Procedure to submit now**

To submit the batch generation job immediately, enter Yes in the Submit Job Now? field.

**Procedure to submit later**

To submit the batch generation job later, first make sure you have saved the JCL in a dataset as described in “Scheduling the Batch Job” on page 195, then enter No in the Submit Job Now? field.
Interpreting a Typical Printed Report

Printed workload and resource reports are provided in two formats: bar graphs and tabular reports.

### USE this format

- **bar graphs**: summary reports
- **tabular format**: detail and exception reports

### FOR these reports...

- a legend at the bottom of the report to define the symbols used to indicate the condition of the workload or resource.
- information presented in columns of data. A horizontal scale usually accompanies each row of data to graphically represent the numeric values in the row.

### TO generate...

<table>
<thead>
<tr>
<th>Interval</th>
<th>Avg Time</th>
<th>% of Total</th>
<th>Average</th>
<th>Avg Time</th>
<th>% of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Warning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example: interpreting a typical detail report

The following example illustrates the structure of a typical detail report.

---

**Average**                                 **Avg Time**   **% of Total**

<table>
<thead>
<tr>
<th>Interval</th>
<th>Resp Time</th>
<th>Main Execution State</th>
<th>In State</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:00-09:00</td>
<td>0.82 sec</td>
<td>DISK OMON29 312 Act</td>
<td>0.28 sec</td>
<td>34.5</td>
</tr>
<tr>
<td>09:00-10:00</td>
<td>1.82 sec</td>
<td>Waiting for CPU</td>
<td>0.46 sec</td>
<td>24.5</td>
</tr>
<tr>
<td>10:00-11:00</td>
<td>0.93 sec</td>
<td>DISK OMON29 312 Que</td>
<td>0.30 sec</td>
<td>32.1</td>
</tr>
<tr>
<td>11:00-12:00</td>
<td>0.85 sec</td>
<td>DISK OMON29 312 Que</td>
<td>0.29 sec</td>
<td>34.5</td>
</tr>
<tr>
<td>12:00-13:00</td>
<td>1.52 sec</td>
<td>DISK OMON29 312 Act</td>
<td>0.17 sec</td>
<td>11.5</td>
</tr>
<tr>
<td>13:00-14:00</td>
<td>0.88 sec</td>
<td>Swapped: Unilateral</td>
<td>0.22 sec</td>
<td>24.5</td>
</tr>
<tr>
<td>14:00-15:00</td>
<td>1.87 sec</td>
<td>Private Page-in</td>
<td>0.42 sec</td>
<td>22.5</td>
</tr>
<tr>
<td>15:00-16:00</td>
<td>0.96 sec</td>
<td>DISK OMON29 312 Act</td>
<td>0.14 sec</td>
<td>14.5</td>
</tr>
<tr>
<td>16:00-17:00</td>
<td>1.05 sec</td>
<td>DISK OMON29 312 Que</td>
<td>0.26 sec</td>
<td>24.5</td>
</tr>
<tr>
<td>17:00-18:00</td>
<td>0.94 sec</td>
<td>DISK OMON29 312 Act</td>
<td>0.32 sec</td>
<td>34.5</td>
</tr>
</tbody>
</table>

**Bottleneck Summary**

<table>
<thead>
<tr>
<th>Interval</th>
<th>Avg Time</th>
<th>% of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:00-18:00</td>
<td>1.20 sec</td>
<td>100.0</td>
</tr>
</tbody>
</table>

---
Interpreting report contents

A typical detail or exception report for workloads and resources is divided into sections that describe the expected service level and a breakdown of activity in intervals of one hour or less. In the previous figure, the workload detail report contains the following information:

- service information that compares the actual service delivered to the user-defined expected service level
- a breakdown of activity at one-hour intervals
- a summary of activity that combines all reported intervals

Interpreting the horizontal scale

The scale to the right of the data rows of the report provides a graphical representation of the average response time, and the average main execution state time during each interval, plotted in units of seconds. A vertical dotted line through the graph marks the point that represents the expected service level. The column heading includes a legend that describes how to interpret the graphical symbols.

Note: Some printed reports (not shown here) feature elapsed time rather than response time. Elapsed time is always reported in minutes.

Source of workload service levels

Workload service levels are reported on workload reports as either elapsed time or response time, depending on the type of workload reported. These service levels are defined as performance group thresholds on the Options pull-down; to check your current threshold settings for performance groups, enter fast path `otp`.

Source of resource service levels

Resource utilization service levels are defined as resource, DASD, and domain thresholds on the Options pull-down. To check your current threshold settings, enter fast path `otu` (for resource utilization thresholds), `otd` (for DASD thresholds), and `otm` (for domain thresholds).

Interpreting report columns
For more information that will help you interpret the columns on these printed reports, see the online helps for each report type and the data dictionary in the *EPILOG for MVS Command Language Reference Manual*.

**Watching Critical Workloads over Time**

**Introduction**
The Workload Profiling Facility (WPF) gathers and averages performance data for selected workloads and constructs a profile that describes “normal” performance for each workload. WPF then allows you to compare the workload’s current performance against its profile.

**MVS/SP 5 and WPF**
WPF gathers IPS-based performance data. It does not gather performance data for WLM-based objects (service classes and report classes) when MVS is running in goal mode.

**Before using WPF**
Before you use WPF, make sure a VSAM dataset has been defined and initialized for the Profile datastore. The Profile datastore is set up during installation. Instructions for its maintenance are contained in the *OMEGAMON II for MVS Configuration and Customization Guide*.
Identifying Critical Workloads for Profiling

Purpose of WPF
The Workload Profiling Facility (WPF) creates profiles of averaged historical performance data using the workloads and selection criteria that you specify. WPF then saves the information so you can later use it to create reports and make comparisons of past and present performance.

Overview of WPF
The following table shows the steps to take to use WPF:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | PROFILE | Create a profile of a workload’s past performance using selected data from the EPILOG datastore.  
|      |         | Save the results in the Profile datastore. |
| 2    | DISPLAY | Create a Profile Report using the EPILOG reporter that shows the workload’s typical performance. |
| 3    | COMPARE | Create a Profile/Workload Comparison Report that measures a workload’s recent degradation against its profile. |
| 4    | SETP    | Set criteria for selecting the profile used with the COMPARE command. |

Which workloads should be profiled?
Although you can use the WPF to profile any workload on your system (batch jobs, TSO sessions, and/or started tasks), it is especially useful for tracking the performance of critical workloads.

Identifying critical workloads
You can tell that a workload is critical if it must be completed within a prescribed time. The following are examples of critical workloads:

- any workload that must run within certain time constraints. This would include, for example, a stream of batch jobs that must complete before CICS can be brought up in the morning.
- a group of TSO users with response time service-level objectives. This would include, for example, a performance group consisting of TSO users that requires a response time below a certain limit.

Defining Meaningful Workload Profiles

Introduction
Not only must you identify critical workloads for your profiles, but you must also select records that will produce meaningful profiles for these workloads.
**Careful selection of records**

Because a profile represents a workload’s average performance, the records you select to generate the profile must be typical of the workload’s performance.

**Example of a meaningful profile**

Suppose you want a profile of your site’s daily SMF analysis job for the past two weeks. You select elapsed-time records that are all between one minute (1:00) and one minute thirty seconds (1:30). Since these records are representative of the job’s performance, a profile based on this sample will be meaningful.

**Examples of skewed profiles**

- Using the same example, this time you include in your sample two records in which the SMF analysis job went into a loop. These records may show very high elapsed times that would skew the average upwards and result in a skewed profile.

- Conversely, using the same example again, you include in your sample a record in which the job was cancelled. The short elapsed time of the record would skew the average downward and once again would produce a skewed profile.

**How much difference can it make?**

The following data shows how excluding three unconventional records with outlying values makes the profile of the SMF analysis job more meaningful:

<table>
<thead>
<tr>
<th>JOB ELAPSED TIME (in minutes)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MON</td>
<td>0:38M</td>
</tr>
<tr>
<td>TUE</td>
<td>1:12M</td>
</tr>
<tr>
<td>WED</td>
<td>1:11M</td>
</tr>
<tr>
<td>THU</td>
<td>1:10M</td>
</tr>
<tr>
<td>FRI</td>
<td>1:14M</td>
</tr>
<tr>
<td>SAT</td>
<td>1:09M</td>
</tr>
<tr>
<td>MON</td>
<td>1:02M</td>
</tr>
<tr>
<td>TUE</td>
<td>1:33M</td>
</tr>
<tr>
<td>WED</td>
<td>1:32M</td>
</tr>
<tr>
<td>THU</td>
<td>2:04M</td>
</tr>
<tr>
<td>FRI</td>
<td>7:01M</td>
</tr>
</tbody>
</table>

- As shown in the left column, accepting all records skews the average elapsed time.

- As shown in the right column, excluding the three outlying values (0:38M, 2:04M, and 7:01M) from the sample produces a more meaningful average elapsed time for the profile.

**Creating Profiles for Workloads**

**Introduction**
Using the PROFILE command, you can tell WPF which records to use to create your profiles. Before you use the PROFILE command, make sure that:

- data is in the EPILOG datastore
- a VSAM dataset has been defined and initialized for the Profile datastore

**Using the PROFILE command**

You create profile records in batch mode using the sample program supplied in `rholev.RKANSAM(KEPPROFJ)`.

*Note:* `rholev` refers to the high-level qualifier for the dataset. To find out what the high-level qualifier is at your site, check with the person who installed OMEGAMON II.

Follow the steps below to complete the procedure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Locate the sample program in <code>rholev.RKANSAM</code> and copy it into another member so the original remains unchanged.</td>
</tr>
<tr>
<td>2</td>
<td>In the copy, modify the PROFILE command.&lt;br&gt;&lt;br&gt;<strong>Note:</strong> For complete information about the PROFILE command and its keywords, see the <em>EPILOG for MVS Command Language Reference Manual</em>.</td>
</tr>
<tr>
<td>3</td>
<td>Submit the job.&lt;br&gt;&lt;br&gt;<strong>Result:</strong> WPF selects the specified data from the historical datastore, creates the profile, and saves it in the Profile datastore.</td>
</tr>
</tbody>
</table>

**Building PROFILE commands**

The examples that follow this discussion illustrate how to build PROFILE commands that create:

- standard (most basic) profiles
- profiles that exclude atypical records
- multiple profiles

**Example of creating a standard profile**

This PROFILE command creates a standard profile and does not filter out any records. As always, a hyphen (-) is used at the end of a line to indicate that the command continues on the next line.

```
PROFILE PGN(2) -
    PNAME('TSO USERS') -
    LASTWEEK
```
Your profile, named TSO USERS, contains the average degradation data of all the historical datastore records for performance group 2 from the previous week, regardless of response time.

**Example of creating a profile that excludes atypical records**

By including the keyword OUTLIER in the PROFILE command, you create a profile that excludes degradation records from the profile with elapsed times that are not representative of that sample.

```
PROFILE JOB(PAYROLL) -
  PNAME('PRIME SHIFT') -
  LASTWEEK BAND STIME(0900) ETIME(1700) -
  SIF(ELAPSED(>30m)) -
  OUTLIER(3)
```

This profile command selects jobs named PAYROLL that ran last week between 9:00 AM and 5:00 PM with an elapsed time greater than 30 minutes, and then excludes from the sample any outlying records with elapsed times that lie 3 levels away from the most typical values.

The resulting profile, named PRIME SHIFT, contains the average degradation data for these records, whose elapsed times are representative for the “long-running PAYROLL jobs” workload.

**Creating multiple profiles**

You can create multiple profiles with one PROFILE command by:

- specifying two or more workload values of the same type
- specifying two or more system IDs
- including an asterisk (*) for a given workload

**Example of specifying two workloads of the same type**

By specifying two workloads of the same type on one PROFILE command, you create two separate profiles.

```
PROFILE PGN(2,12) -
  PNAME(TSO) -
  LASTWEEK BAND STIME(0900) ETIME(1700) -
  SIF(RESPONSE(>2s)) -
  OUTLIER(3)
```

This command creates two separate profiles, one for performance group 2 and the other for performance group 12 (workloads of the same type).
This profile definition selects performance group 2 and performance group 12 that ran last week between 9:00 AM and 5:00 PM with a response time greater than 2 seconds, and then excludes from the sample any outlying records with response times that lie 3 levels away from the most typical values.

The resulting profiles, named TSO, contain the average degradation data for the records whose response times are representative for the workloads.

Since these profiles have the same name (TSO), when using them on the DISPLAY and COMPARE commands, you must always include the workload name for each profile to tell them apart.

**Example of specifying two or more system IDs**

By specifying two or more system IDs on one PROFILE command, you create separate profiles for the selected workload under each system.

```
PROFILE PGN(2) -
    PNAME('TSO DEV') -
    SYSID(SYSA,SYSB) -
    THISMONTH -
```

This command creates a profile named TSO DEV for performance group 2 under SMF system ID SYSA and a profile named TSO DEV for performance group 2 under SMF system ID SYSB.

Each profile contains the workload’s average degradation data for the current month. WPF excludes from the data sample any records that fall outside the third outlier interval.

**Note:** If you omit the system ID, WPF combines the average degradation data for all workloads that meet the other selection criteria and generates only one profile.

**Example of creating profiles using a mask**

Use a mask(*) for a workload to create profiles for an entire workload type.

```
PROFILE TSO(*) -
    PNAME('PRIME SHIFT') -
    LASTWEEK BAND STIME(0900) ETIME(1700) -
    OUTLIER(3)
```

Placing an asterisk next to TSO masks the TSO ID and has the same result as specifying an individual PROFILE command for each TSO ID on the system.

This profile definition selects all TSO workloads that ran last week between 9:00 AM and 5:00 PM and then excludes from the sample any record that falls outside the third level of outliers.
Each resulting profile, named PRIME SHIFT, contains the average degradation data for the representative records.

Since these profiles have the same name (PRIME SHIFT), when using them on the DISPLAY and COMPARE commands, you must always include the workload name for each profile to tell them apart.

Testing Data before Writing It to a Profile Datastore

Introduction
By using the TEST keyword in your PROFILE command, you can instruct WPF not to write the profile it has generated to the Profile datastore. This allows you to experiment with different profile definitions without wasting datastore space.

TEST helps you with OUTLIER
When you are creating the profile for the first time, the TEST keyword enables you to see the results that will be profiled before actually creating the profile. This way you can experiment with OUTLIER values and determine which one generates the most meaningful profile without wasting datastore space.

PLOT helps you with TEST
When using the TEST keyword, you should also specify the PLOT keyword or leave it as the default. The PLOT keyword will generate a graph of the workload’s elapsed times and show the different ranges of outliers generated by your test profile.

Note: If you do not want a graph generated, specify the NOPLOT keyword.

Testing and graphing your profile
Perform the following steps to test and graph your profile.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Locate the sample program in rhilev.RKANSAM and copy it into another member so the original remains unchanged.</td>
</tr>
</tbody>
</table>
| 2    | In the copy, enter a PROFILE command that specifies the TEST keyword and leaves PLOT as the default.  
Example:  
PROFILE PERFGROUP(2) PNAME PLOT - STIME(0900) ETIME(1700) DAYOFWK(WKDAY) TEST |
| 3    | Submit the job.  
Result: WPF generates a batch report that graphs (PLOT) the response times and their frequency but does not save (TEST) the profile to the Profile datastore. |
| 4    | Examine the graph of the sample to determine the range of OUTLIER values. |
Displaying Profiles

Introduction
Using the DISPLAY command, you can create detailed online displays or batch reports for profiles.

Before you begin
This information assumes that you have already created profiles using the PROFILE command and have saved them in the Profile datastore.

Using DISPLAY to generate online reports
The examples that follow this discussion illustrate how to build DISPLAY commands that can be used interactively to create online profile displays or in batch mode to generate Profile reports.

To see the display online, enter the DISPLAY command on the EPILOG command line (enter fast path ge on any OMEGAMON II panel to display the EPILOG command line).

Invoking the batch reporter
To invoke the reporter in batch mode, add the DISPLAY command to the sample program supplied in rhilev.RKANSAM(KEPPROC).

Example of creating a profile display for a specific time interval
By specifying the start and end dates in your DISPLAY command, you create a profile display for a specific time interval.

```
DISPLAY JOB(DEV*) -
   PNAME('PRIME SHIFT') -
   SDATE(03/01/94) -
   EDATE(03/31/94)
```

The resulting display shows the profile named PRIME SHIFT for every job with a name that begins with DEV that ran between March 1 and March 31 of 1994.
Example of creating a profile display for specific system IDs
By specifying two system IDs in your DISPLAY command, you create a display showing profiles for both systems.

```sql
DISPLAY JOB(PRSNL) -
       PNAME('PRIME SHIFT') -
       SYSID(SYSA,SYSB) -
       LASTMONTH
```

The resulting display shows profiles for JOB(PRSNL) that ran for system A and system B. The selected profiles are named PRIME SHIFT and contain average degradation data from the previous month for the job named PRSNL.

**Note:**
- This example assumes that the Profile datastore contains some profiles with system ID SYSA and system ID SYSB.
- Omitting system ID displays a profile that meets the other selection criteria for each ID under which a profile was created.

Example of creating a profile display for an entire workload type
By placing an asterisk next to the workload and the profile name in your DISPLAY command, you create a display for an entire workload type.

```sql
DISPLAY STC(*) -
       PNAME(*)
```

The resulting display shows all profiles for all started tasks.

Example of creating a profile display with device detail
By including the IODEVICE keyword in your DISPLAY command, you create a display that includes individual unit addresses for any device that is associated with a wait reason.

```sql
DISPLAY JOB(ADMIN01) -
       PNAME(PAYROLL) -
       SDATE(03/01/94) EDATE(03/31/94) -
       IODEVICE
```
The resulting display shows any profiles named PAYROLL that were created for ADMIN01 jobs for any period between March 1 and March 31, 1994.

Interpreting Profile Reports

Introduction
The DISPLAY command can be used to generate the following types of profile reports:
- Standard Profile Report
- Profile Report with Device Detail

The only difference between the two batch reports is that the Profile Report with Device Detail displays the unit addresses for the I/O devices listed under wait reasons.

Invoking the batch reporter for the Standard Profile Report
Suppose you want to create a standard profile report, in batch mode, using this DISPLAY command:

```
DISPLAY JOB(SORTNYP) PROFNAME('PRIME SHIFT') -
LASTWEEK
```

The following procedure is used to invoke the batch reporter.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Locate the sample program in <code>rhilev.RKANSAM.KEPPROC</code> and copy it into another member so the original remains unchanged.</td>
</tr>
<tr>
<td>2</td>
<td>In the copy, enter the DISPLAY command.</td>
</tr>
<tr>
<td>3</td>
<td>Submit the job. Result: The Standard Profile Report is generated.</td>
</tr>
</tbody>
</table>

Looking at a sample Standard Profile Report
The Standard Profile Report looks like this.
Identifying Critical Workloads for Profiling

Interpreting the sample report

You see that the report in the previous example shows your workload/profile name/time specifications, as well as the OUTLIER criteria used to create the profile (line 3). Seventeen historical datastore records were included and one was excluded after OUTLIER processing (line 4).

The report body shows a generic listing of wait reasons. In this case the workload SORTNYP spent the most time waiting on I/O Active followed by Waiting for CPU. You notice that sixteen batch jobs were included in this profile (lower-right-hand corner) and that the productivity index was 22%.

Note: The number of historical datastore (EDS) records included after OUTLIER processing is different from the number of jobs included in the profile because EDS records are kept for each job step.

Looking at a sample Profile Report with Device Detail

If you want to see a profile report with the unit addresses for the I/O devices associated with these wait reasons, you must add the IODEVICE keyword to the DISPLAY command.

DISPLAY with IODEVICE generates the following report:
Identifying Critical Workloads for Profiling

The resulting display is identical to the previous one except for the listing of disk wait reasons by unit address.

Note: Specifying IODEVICE on a DISPLAY command may produce a lengthy report since, in the profiles of some workloads, a number of devices may be associated with the various wait reasons.

Comparing Workloads to their Profiles

Introduction

Using the COMPARE command, you can compare a workload’s recent degradation data with a profile of its past degradation data and produce a display that shows the differences in performance.

Before you begin

Since the data used for comparison reports is taken from the EPILOG datastore (for the workload) and the Profile datastore (for the profile), the information must already be in place in these datastores for the COMPARE command to work.

Using COMPARE to generate online reports

The examples that follow this discussion illustrate how to build COMPARE commands that can be used interactively to create online comparisons or in batch mode to generate comparison reports.

To see the comparison online, enter the COMPARE command on the EPILOG command line (enter fast path ge on any OMEGAMON II panel to display the EPILOG command line).

Note: If a comparison requires processing large amounts of data (for example, a one-month time period specification), you should use batch mode.
**Invoking the batch reporter for comparison reports**

To generate a comparison report in batch mode, add the COMPARE command to the sample program provided in `rhilev.RKANSAM(KEPSRCMP)

**Example of creating a Comparison Report**

This COMPARE command creates a report that compares the recent degradation of a selected workload (in this case, the job named PAYROLL) with its most current profile.

```
COMPARE JOB(PAYROLL) -
  PNAME(PAY) -
  TODAY -
  RIF(ELAPSED(>15m)
```

The RIF keyword tells EPILOG that you only want it to display those historical datastore records with elapsed times exceeding the profiles by 15 minutes.

**Note:** RIF applies only to the historical datastore records for the specified workload and not to the profile.

**Example of creating a Comparison Report for multiple system IDs**

By specifying multiple system IDs in your COMPARE command, you create a report that compares qualifying workload-profile combinations for each of those system IDs.

```
COMPARE JOB(PAYROLL) -
  PNAME(PAY) -
  SYSID(SYSA,SYSB) -
  LASTWEEK -
  STIME(1700) ETIME(0700)
```

In this instance, for both system A and system B, your report compares the historical datastore records for the job PAYROLL from the previous week between 5:00 PM and 7:00 AM against the most recent profile that has the name PAY and that was created for the job PAYROLL. Jobs run on system A are measured against the profile created on system A and jobs run on system B are measured against the profile created on system B.

**Example of creating a Comparison Summary Report showing deviation**

By substituting an asterisk (*) for a job name, setting the PNAME to default, and specifying the SUMMARY keyword in your COMPARE command, you create a summary report that shows workload deviation.

```
COMPARE JOB(*) -
  PNAME(default) -
  TODAY -
  SUMMARY
```
The resulting report displays, in descending order, the workloads that deviated most from their profiles.

Because you have not limited the number of records to be displayed, your report lists up to twenty workloads (the default).

**Setting default values for comparisons**

Using the SETP command, you can compare a workload with a profile other than the most current one. By establishing defaults for the workload, time period, and/or system ID, you are setting the values used by the COMPARE command to select the profile from the Profile datastore.

**Guidelines for building SETP commands**

The SETP command establishes default values for the COMPARE command to use in selecting the profile from the Profile datastore. There are keywords for setting the workload, time period, and system ID.

Because SETP defaults remain in effect until you enter a subsequent SETP command or cancel them with SETP CLEAR, you need to be aware of how multiple SETP commands will be interpreted:

- Until you cancel the default values established by SETP commands, they are combined to create a cumulative default value.
- A workload value replaces a previous workload.
- A time period value replaces a duplicate type.

**Example:** If one SETP command establishes SDATE (6/1) and EDATE (6/30), then a later SETP command establishing PGN(2), STIME(0900), and ETIME(1700) would combine with it.

**Example:** If one SETP command establishes PGN(2), then a later one establishing PGN(12) would override the previous workload value.

**Example:** If one SETP command establishes STIME(0900) then a later SETP command establishing STIME(1200) would override the previous time period default value.

**Note:** For a complete explanation of SETP and its keywords, see the *EPILOG for MVS Command Language Reference Manual.*

**Interpreting Comparison Reports**

**Introduction**

Your COMPARE command creates detailed, easy-to-read reports.
Comparison reports are of two types:

- The Profile/Workload Comparison Report, which shows a workload’s current degradation measured against its past performance
- The Profile/Workload Comparison Summary Report, which lists workloads that show the most deviation from their profiles

**Invoking the batch reporter for a comparison report**

Suppose you want to create a Profile/Workload Comparison Report, in batch mode, using this COMPARE command:

```
COMPARE JOB(MGTMN01) PROFNAME(MGTJOBS) YDAY -
RIF (ELAPSED(>10%))
```

The following procedure is used to invoke the batch reporter.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Locate the sample program in <code>rhilev.RKANSAM(KEPSRCMP)</code> and copy it into another member so the original remains unchanged.</td>
</tr>
<tr>
<td>2</td>
<td>In the copy add the COMPARE command.</td>
</tr>
<tr>
<td>3</td>
<td>Submit the job.</td>
</tr>
</tbody>
</table>

**Result:** The comparison report is generated.

**Looking at a sample Profile/Workload Comparison Report**

The Profile/Workload Comparison Report looks like this.

```
+------------------------+------------------------+------------------------+------------------------+------------------------+
|                        |                        |                        |                        |
| Job = MGMTN01          | Prof Job = MGMTN01     | Profname = MGTJOBS     |                        |
| Prof Period: 00:01 to 12:10 on 04/03/94 |                        |                        | Sysid = SYSA           |
| Wkld Period: 09:29 to 09:30 on 05/05/94 |                        |                        | Sysid = SYSA           |
|                        |                        |                        |                        |
| |                        |                        |                        |                        |
| Wait_Reason____________Profile_Workload|16.59___ 8.29_____0_____ 8.29___16.59|                        |
| Using CPU              2.48 S   0.26 S |                        |                        |
| Waiting for CPU        9.80 S  26.39 S |                        |                        |
| Enqueue                0.95 S  11.73 S |                        |                        |
| I/O Queued             1.19 S   2.93 S |                        |                        |
| ECB Wait               1.43 S   2.93 S |                        |                        |
| Swap Page-In Wait      0.23 S    n/a   |                        |                        |
| I/O Reserved           0.71 S    n/a   |                        |                        |
| SRM Delay (MPL)        1.67 S    n/a   |                        |                        |
| Waiting for MVS Lock   2.15 S    n/a   |                        |                        |
| I/O Active             14.34 S   n/a   |                        |                        |
| Elapsed Time           34.99 S  44.27 S |                        |                        |
| Productivity Index     53%      41%                                           |
+------------------------+------------------------+------------------------+------------------------+------------------------+
```
While the display header shows your workload/profile name/time specifications, the display body compares the workload with its profile with regard to its:

- major wait reasons
- elapsed time
- productivity index
- graph deviation

**Interpreting the sample report**

In the sample Profile/Workload Comparison Report, you see that for the workload MGTMN01:

- **Waiting for CPU** is the wait reason with the highest elapsed time deviation from the profile, with the list continuing in descending order. (**Waiting for CPU** is *always* listed first.)

- Many wait reasons such as **Swap Page-In Wait** are major ones for the profile but not for the workload (indicated with n/a).

- The workload’s elapsed time is 9.28 seconds greater than the profile’s.

- The profile’s productivity index is 12% higher.

**Looking at a sample Profile/Workload Comparison Report with device detail**

Now you want to see the preceding report with the unit addresses for the devices listed under wait reasons.

You specify the IODEVICE keyword when you issue the COMPARE command and this comparison report is generated.
The resulting report looks similar to the previous one, except that disk wait reasons are listed by unit address.

As with the DISPLAY command, specifying IODEVICE on a COMPARE command may produce a rather lengthy report since many devices may be associated with the various wait reasons.

Looking at a sample Profile/Workload Comparison Summary Report showing deviation

Finally, suppose you want a comparison of data in summary form showing the workloads that deviated most from their profiles.

You submit this COMPARE command:

```
COMPARE JOB(*) PROFNAME SUMMARY
```

The result is the following report showing all jobs (indicated with asterisks) with the PNAME default for the specified time period.
Interpreting the sample report

Using workload USER08AA as an example, you see that:

- its elapsed time deviates from its profile elapsed time by 18:31 minutes (that is, its elapsed time was 18:31 minutes longer)

  **Note:** A minus sign (-) indicates that the workload’s response time was less than the profile’s.

- its run time was 19:13 minutes, starting at 9:29 on 04/03/94 as compared to its profile’s run time of 42.01 seconds, starting on 05/11/94 and ending on 05/11/94

Monitoring a Realtime Workload against Its Profile

Introduction

OMEGAMON works with the EPILOG Workload Profiling Facility to monitor the status of workloads against their profiles. It provides monitoring commands and notifies you by means of exception analysis when a workload exceeds its profile.

Before you begin to use the WPF command, you should make sure that the Profile datastore is accessible to OMEGAMON. Also, WPF exception criteria (including baseline and threshold
data) and status alerts must be established in OMEGAMON. See your OMEGAMON II Customizer to verify that this has been done.

- The OMEGAMON commands described in this section are entered below the INFO-line of an OMEGAMON screen. To prepare for entering the commands, enter fast path go on any OMEGAMON II panel, then press PF24.

For complete information on all of the OMEGAMON commands used here, see the OMEGAMON for MVS Command Language Reference Manual.

**Example of the WPF command**
The following examples show you how to start WPF, display WPF status, and stop WPF.

- To activate WPF, enter this command, with your profile dataset in parentheses:

  ```
  WPF START PRDS(USER01.V750.RKM2PRDS)
  ```

  OMEGAMON then displays WPF’s “initial” status as shown below:

  ![WPF initial status](image)

- To display WPF status at any time after this, enter the WPF command with no operand:

  ```
  WPF
  ```

  OMEGAMON confirms WPF’s “active” status showing that WPF is collecting profile data and monitoring workloads against profiles.

  ![WPF active status](image)

- To stop WPF and release associated datasets, enter this command:

  ```
  WPF STOP
  ```

  OMEGAMON then displays WPF’s “inactive” status as shown below.
Identifying Critical Workloads for Profiling

Displaying WPF status and profile listings
Suppose you want to display WPF status and also list the profiles in use for all started tasks.

Enter the following command:

```
dWPF STC(*)
```

In response to this command, OMEGAMON shows WPF as active and lists the names of profiles currently in use, as shown in the following figure.

```
+ PRDS(USER01.V750.RKM2PRDS) status is inactive
```

Displaying WPF status and workload information
Now suppose you want to display WPF status, along with information on workloads for all started tasks.

Enter the following command:

```
jWPF STC(*)
```

In response to this command, OMEGAMON shows WPF as active and provides the information you requested on workloads, as shown in the following figure.
Setting exceptions

WPF generates an exception whenever the elapsed or response time of the workload exceeds that of the profile by a given percent. Information on these exceptions is provided in the OMEGAMON for MVS Command Language Reference Manual under the XACB command.
Introduction

The flexibility of OMEGAMON II allows you to extract historical data and then manipulate that data to create customized statistical or graphic reports that meet the needs of your organization. The methods available to you are:

- the OBTAIN command, which enables you to export data to other applications
- the interfaces to SAS™ and SLR, which produce graphical reports from the data you have exported. For information on the SLR interface, see the OMEGACENTER Reporting with SLR Reference Manual.

Chapter Contents

Creating a Historical Output File .................................................. 220
Producing SAS Graphical Reports ................................................. 230
Creating a Historical Output File

The OBTAIN command exports data from an EPILOG datastore to a sequential output file where it can be used by mainframe and PC application packages. This section shows how to create your own historical output file using the OBTAIN command.

In this section

In this section we cover the following topics.
- “Preparing to Use the Sample Proc or CLIST” on page 220
- “Defining the Data to be Exported” on page 221
- “Interpreting EBCDIC Columnar Output Files: FORMAT(COL)” on page 223
- “Interpreting Binary Output Files: FORMAT(INT)” on page 225
- “Interpreting EBCDIC Columnar Output Files with Delimiters: FORMAT(PC)” on page 227

Overview of how to create your own historical output file

These are the steps in exporting data using the OBTAIN command.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Edit the sample proc or CLIST, setting the system parameters for your installation.</td>
</tr>
<tr>
<td>2</td>
<td>Use the OBTAIN and SET commands to describe the data that you want to export.</td>
</tr>
<tr>
<td>3</td>
<td>Place your OBTAIN and SET commands in a dataset.</td>
</tr>
<tr>
<td>4</td>
<td>Submit the job (to run the proc), or execute the CLIST.</td>
</tr>
</tbody>
</table>

Preparing to Use the Sample Proc or CLIST

To export data with the OBTAIN command, you can either submit a batch job or execute a CLIST.

Sample files to help you run the OBTAIN command

Two sample files are included on the product tape to help you run the OBTAIN command.

<table>
<thead>
<tr>
<th>IF you want to use...</th>
<th>THEN look at the sample file found in....</th>
</tr>
</thead>
<tbody>
<tr>
<td>a sample proc</td>
<td>rhilev.RKANSAM(KEPOBT)</td>
</tr>
<tr>
<td>a sample CLIST</td>
<td>rhilev.RKANCLI(KEPOBTC)</td>
</tr>
</tbody>
</table>

Modifying the sample files
The proc or CLIST must be modified to meet your needs.

<table>
<thead>
<tr>
<th>IF you need to change the definition of...</th>
<th>THEN modify DD statement...</th>
</tr>
</thead>
<tbody>
<tr>
<td>the output file</td>
<td>RKM2OUTD, as explained in “Defining the output file(s)” on page 221</td>
</tr>
<tr>
<td>the data to be exported</td>
<td>SYSIN, as explained in “Defining the Data to be Exported” on page 221. This is done with the OBTAIN command.</td>
</tr>
<tr>
<td>the datastore from which the data is to be extracted</td>
<td>RKANPAR, which defines the library that contains the KEPEDS dataset. KEPEDS lists the datastores that can serve as the source of your data.</td>
</tr>
</tbody>
</table>

### Defining the output file(s)

The output file(s) for the data that OBTAIN exports are defined as follows.

<table>
<thead>
<tr>
<th>IF you want...</th>
<th>THEN define...</th>
</tr>
</thead>
<tbody>
<tr>
<td>your entire command stream to write to a single output file</td>
<td>an available dataset in the RKM2OUTD DD statement. This is the default method. Alternatively, you may define a different dataset using a different DD statement. In this case the DD statement must correspond to the DD name in the OUTFILE keyword of the OBTAIN command.</td>
</tr>
<tr>
<td>to place data in multiple output files</td>
<td>several available datasets by defining several DD statements. Each DD statement must correspond to a DD name in the OUTFILE keyword of an OBTAIN command in the OBTAIN command stream. You may then use multiple OBTAIN commands in one command stream to write to multiple output files.</td>
</tr>
</tbody>
</table>

### Defining the Data to be Exported

Each OBTAIN command extracts requested information from EPILOG datastores and writes that information to a sequential output file that contains variable length records.

### What data can you extract?

You can extract data elements detailing channel, CPU, and DASD resource utilization, as well as (for example) SRM domain activity and I/O queuing information. For a complete description of the available data elements, see the discussion in the data dictionary portion of the *EPILOG for MVS Command Language Reference Manual*.

### What data can’t you extract?

Under MVS/SP 5, the historical collector gathers WLM-based data when MVS is running in goal mode, and IPS-based data when MVS is running in compatibility mode. You cannot extract IPS-based data elements from a datastore that was filled while MVS was running in goal mode.
Creating a Historical Output File

Example of the OBTAIN command
The following OBTAIN command could be used to export information on the minimum, maximum, and average number of batch jobs running last week to Lotus® 1-2-3®.

```
OBTAIN RCPU FORMAT(COL) -
    ELEMENTS(BATMINAS, BATMAXAS, BATAVGAS) -
    OUTFILE(MYDATA) APPEND LASTWEEK
```

The APPEND keyword ensures that this OBTAIN command will not overwrite anything already in MYDATA. For complete information on all the OBTAIN keywords, see the EPILOG for MVS Command Language Reference Manual.

Defining global parameters using the SET command
When planning to code a command stream with several similar OBTAIN commands, you may want to consider using the SET command to establish global default parameters for the entire job. For example, if you wanted to gather a variety of information, all in FORMAT(COL), you could precede your first OBTAIN command with this command:

```
SET FORMAT(COL)
```

You could then code as many OBTAIN commands as you like in that job without having to include FORMAT(COL) on any of them.

For complete information on all of the keywords used with SET, see the EPILOG for MVS Command Language Reference Manual.

Types of output formats
The format of the output file is controlled by the FORMAT keyword on an OBTAIN or SET command in the command file.

<table>
<thead>
<tr>
<th>IF you need output in...</th>
<th>THEN use this keyword...</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBCDIC columnar format for possible conversion to ASCII</td>
<td>FORMAT(COL)</td>
</tr>
<tr>
<td>EBCDIC columnar format with delimiters for possible conversion to ASCII</td>
<td>FORMAT(PC)</td>
</tr>
<tr>
<td>binary format</td>
<td>FORMAT(INT)</td>
</tr>
</tbody>
</table>

Pointing to the OBTAIN commands
The RKM2IN DD statement defines where to find the OBTAIN commands. Use

```
//RKM2IN DD *
```

if the OBTAIN commands immediately follow this DD statement; or

```
//RKM2IN DD dataset
```
where **dataset** is the name of the dataset in which you put your OBTAIN commands.

**Submitting your batch job or running your CLIST**

How you submit your batch job or run your CLIST depends on your particular site. For guidance, look in the appropriate sample file. Because it is difficult to estimate the number of observation records to expect, we recommend that you use generous secondary space allocations for your output file.

**Interpreting your output**

As previously explained, the format of your output file is controlled by the FORMAT keyword. For help in interpreting your output, please refer to the material indicated in the following table.

<table>
<thead>
<tr>
<th>IF you used...</th>
<th>THEN see...</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMAT(COL)</td>
<td>“Interpreting EBCDIC Columnar Output Files: FORMAT(COL)” on page 223</td>
</tr>
<tr>
<td>FORMAT(INT)</td>
<td>“Interpreting Binary Output Files: FORMAT(INT)” on page 225</td>
</tr>
<tr>
<td>FORMAT(PC)</td>
<td>“Interpreting EBCDIC Columnar Output Files with Delimiters: FORMAT(PC)” on page 227</td>
</tr>
</tbody>
</table>

**Interpreting EBCDIC Columnar Output Files: FORMAT(COL)**

Output files created with the FORMAT(COL) keyword contain EBCDIC output in columns. Generating your output in this format is useful if you’re using the output files from OBTAIN as input files to mainframe graphics or statistics packages, or to certain PC software packages.

**FORMAT(COL) record types**

The FORMAT(COL) output file contains two record types.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>What It Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute record</td>
<td>Description of the contents of the observation records</td>
</tr>
<tr>
<td>Observation record</td>
<td>Element values selected from the datastore</td>
</tr>
</tbody>
</table>

**Attribute record structure for FORMAT(COL)**

Your output file will contain one attribute record for each OBTAIN command in the command stream. The record consists of a list of descriptions, one description for each element you requested. Each description contains the fields shown in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>element name, in character representation, preceded by an asterisk (*)</td>
</tr>
</tbody>
</table>
Creating a Historical Output File

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>offset (position of the field from the beginning of the record) in integer representation</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>length of the element in bytes, in integer representation</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>data type (N=numeric signed integer, C=character, (H=)hexadecimal, or (P=)packed decimal)</td>
</tr>
</tbody>
</table>

The asterisk acts as a delimiter, flagging the beginning of the description of the next data element.

**Observation records for FORMAT(COL)**

The number of observation records depends on what workload or resource keyword you’re using. A workload keyword like JOBNAME, for example, may generate several observation records (one for each wait reason) per collection interval. You can reduce the number of observation records with the COMBINE keyword, which acts to combine data across collection intervals. Other keywords that may affect the number of observation records are PLOTMIN, REPORTIF, and SELECTIF.

Each observation record contains a value field for each data element selected by the user, in the order listed on the attribute record.

**Command sequence example using FORMAT(COL)**

The following command stream generates one attribute record and eight observation records.

```plaintext
SET FORMAT(COL)
SET OUTFILE(RINFOUT) REPLACE
SET ELEMENTS(SMFID)
OBT RINF STIME(9) ETIME(11) SDATE(8/2) EDATE(8/2) -
   ELEMENTS(IPS,ICS,RMF)
END
```

The SET command in this example specifies the default format of all output files created by subsequent OBTAIN commands in this job.

**Output file example using FORMAT(COL)**

```
*SDATE   00 008 N *STIME    08 008 C *SMFID   10 004 C *IPS    14 002 C
*ICS     16 002 C *RMF    18 020 C
008921409:00:04SYSAAAAARMF 4.1.1
008921409:15:02SYSAAAAARMF 4.1.1
008921409:30:04SYSAAAAARMF 4.1.1
008921409:45:01SYSAAAAARMF 4.1.1
008921410:00:00SYSAAAAARMF 4.1.1
008921410:15:01SYSAAAAARMF 4.1.1
008921410:30:02SYSAAAAARMF 4.1.1
008921410:45:01SYSAAAAARMF 4.1.1
```

The output file pictured above will be written to the dataset defined by the RINFOUT DD statement, as specified on the OUTFILE keyword.
The records displayed in the figure are in the same format as they would appear if you listed the file at a terminal or printer.

Note that since the default sampling interval is 15 minutes and the requested reporting interval is 2 hours (9 to 11), eight observation records are produced.

**Interpreting Binary Output Files: FORMAT(INT)**

Output files created with the FORMAT(INT) keyword contain binary data. Generating your output in this format is useful if you are writing your own programs to read the output file.

If you were to list the contents of an OBTAIN output file generated with FORMAT(INT), much of it would be non-readable, binary data.

**FORMAT(INT) record types**

The FORMAT(INT) output file contains the following three record types.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>What It Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment record</td>
<td>Text of the OBTAIN commands in the command stream</td>
</tr>
<tr>
<td>Attribute record</td>
<td>Description of the contents of the observation records</td>
</tr>
<tr>
<td>Observation record</td>
<td>Element values selected from the datastore</td>
</tr>
</tbody>
</table>

**Comment record structure for FORMAT(INT)**

Your output file will contain one or more comment records for each OBTAIN command in the command stream. The following table describes the fields in each record:

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>Julian date on which the data was extracted. Shown in packed decimal format.</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>time stamp showing the time of day that the data was extracted.</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>record-type indicator of *, indicating that this is a comment record.</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>reserved field</td>
</tr>
<tr>
<td>11</td>
<td>variable</td>
<td>text of the OBTAIN command. The first record starts with SDATE and STIME, followed by elements listed in the order you requested them. This field also contains the current settings of any valid keywords that are used for data filtering (such as DATE, TIME, BAND, RANGE, RIF, or SIF). For complete information on all of the keywords, refer to the EPILOG for MVS Command Language Reference Manual.</td>
</tr>
</tbody>
</table>

**Attribute record structure for FORMAT(INT)**
Your output file will contain one attribute record for each OBTAIN command in the command stream. The record contains the fields described in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>Julian date on which the data was extracted, shown in packed decimal representation.</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>time stamp showing the time of day that the data was extracted.</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>record-type indicator of A, indicating that this is an attribute record.</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>reserved field</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>the number of elements being OBTAINed (that is, the number of value fields that will appear in each observation record), shown in integer representation.</td>
</tr>
<tr>
<td>13</td>
<td>15 per data element</td>
<td>the element definition of each element you requested, shown in character representation. The requested data elements are listed in the order in which you requested them.</td>
</tr>
</tbody>
</table>

As explained in the last row of the previous table, the element definition section of the attribute record starts in column 13 and includes 15 bytes of explanatory data for each data element requested. These 15 bytes are divided into the fields shown in the following table.

<table>
<thead>
<tr>
<th>Length</th>
<th>Field Description</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>element name</td>
<td>character</td>
</tr>
<tr>
<td>1</td>
<td>data type (I=numeric signed integer, U=unsigned integer, P=packed decimal, C=character, D=floating-point, X=hexadecimal, and T=time)</td>
<td>character</td>
</tr>
<tr>
<td>1</td>
<td>element length, in bytes</td>
<td>integer</td>
</tr>
<tr>
<td>1</td>
<td>decimal point displacement (number of positions to the right of the decimal; for example, CPU seconds may be shown to the nearest tenth of a second for a decimal point displacement of 1)</td>
<td>integer</td>
</tr>
<tr>
<td>4</td>
<td>field offset (location of the field from the beginning of the observation record, in bytes)</td>
<td>integer</td>
</tr>
</tbody>
</table>

**Observation record structure for FORMAT(INT)**

The number of observation records depends on what workload or resource keyword you are using. A workload keyword like JOBNAME, for example, may generate several observation records (one for each wait reason) per collection interval. You can reduce the number of observation records with the COMBINE keyword, which acts to combine data across
collection intervals. Other keywords that may affect the number of observation records are PLOTMIN, REPORTIF, and SELECTIF.

Every observation record contains the fields shown in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>Julian date on which the data was extracted, in packed decimal representation.</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>time stamp showing the time of day that the data was extracted.</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>record-type indicator of O, indicating that this is an observation record.</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>reserved field</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>the field SDATE, which shows the Julian date on which the data was recorded, in packed decimal representation.</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>the field STIME, which shows the start time of the interval for which the data was recorded.</td>
</tr>
<tr>
<td>19</td>
<td>variable</td>
<td>value field for each element you selected, in the order listed in the attribute record.</td>
</tr>
</tbody>
</table>

**Command sequence example using FORMAT(INT)**

The following sample command sequence generated three comment records, one attribute record, and eight observation records.

```
SET FORMAT(INT)
.* SET OUTFILE(RINF) REPLACE
.* SET ELEMENTS(SMFID)
OBT RINF STIME(9) ETIME(11) SDATE(8/2) EDATE(8/2) -
  ELEMENTS(IPS,ICS,RMF)
END
```

The SET command in this example specifies the default format of all output files created by subsequent OBTAIN commands in this job.

**The output file for FORMAT(INT)**

If you were to list the contents of an OBTAIN output file generated with FORMAT(INT), much of the data would be non-readable, binary data. It is for this reason that we do not provide a display of the output file generated with the sample FORMAT(INT) command stream.

**Interpreting EBCDIC Columnar Output Files with Delimiters: FORMAT(PC)**

Output files created with the FORMAT(PC) keyword contain EBCDIC output with delimiters. Generating your output in this format is useful if you are using the output files from OBTAIN as input files to a PC software package that requires delimiter characters.
Creating a Historical Output File

**FORMAT(PC) record types**

The FORMAT(PC) output file contains two record types.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>What It Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute record</td>
<td>Description of the contents of the observation records</td>
</tr>
<tr>
<td>Observation record</td>
<td>Element values selected from the datastore</td>
</tr>
</tbody>
</table>

**Attribute record example for FORMAT(PC)**

Your output file will contain one attribute record for each OBTAIN command in the command stream. An example of an attribute record is shown below.

```
"SDATE  " "STIME  " "SMFID  " "IPS  " "ICS  " "RMF  "
```

The record contains the element names that have been selected, delimited by double quotes.

**Observation record structure for FORMAT(PC)**

The number of observation records depends on what workload or resource keyword you’re using. A workload keyword like JOBNAME, for example, may generate several observation records (one for each wait reason) per collection interval. You can reduce the number of observation records with the COMBINE keyword, which acts to combine data across collection intervals. Other keywords that may affect the number of observation records are PLOTMIN, REPORTIF, and SELECTIF.

Every observation record contains the fields shown in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>the field SDATE, which shows the Julian date on which the data was recorded. Shown in packed decimal format.</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>the field STIME, which shows the start time of the interval for which the data was recorded.</td>
</tr>
<tr>
<td>9</td>
<td>variable</td>
<td>value field for each element you selected, in the order listed on the attribute record. If the value is in character, time, or hexadecimal format, it will be enclosed in double quotes.</td>
</tr>
</tbody>
</table>

**Command sequence example using FORMAT(PC)**

The following command example generates one attribute record and nine observation records.

```
SET FORMAT(PC)
OBT RINF STIME(8) ETIME(17) SDATE(8/2) EDATE(8/2) -
    COMBINE(1H) ELEMENTS(IPS,ICS,RMF)
END
```
The SET command in this example specifies the default format of all output files created by subsequent OBTAIN commands in this job.

**Output file example using FORMAT(PC)**

<table>
<thead>
<tr>
<th>SDATE</th>
<th>STIME</th>
<th>SMFID</th>
<th>IPS</th>
<th>ICS</th>
<th>RMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0089214</td>
<td>08:00:01</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
<tr>
<td>0089214</td>
<td>09:00:01</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
<tr>
<td>0089214</td>
<td>10:00:01</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
<tr>
<td>0089214</td>
<td>11:00:01</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
<tr>
<td>0089214</td>
<td>12:00:04</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
<tr>
<td>0089214</td>
<td>13:00:02</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
<tr>
<td>0089214</td>
<td>14:04:28</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
<tr>
<td>0089214</td>
<td>15:00:01</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
<tr>
<td>0089214</td>
<td>16:00:01</td>
<td>SYSA</td>
<td>AA</td>
<td>AA</td>
<td>RMF 4.1.1</td>
</tr>
</tbody>
</table>

The records displayed in the figure are in the same readable format as they would appear if you listed the file at a terminal or printer.

Since the requested combine period was one hour on this OBTAIN command and the requested reporting interval is nine hours, nine observation records are produced.
Producing SAS Graphical Reports

OMEGAMON II provides an interface to the SAS graphics component. SAS converts technically complex data about system performance and resource utilization into concise, easy-to-read graphs that show you trends at a glance.

You can use these graphs to resolve performance problems, identify present and future resource needs, and ease management reporting.

*Note:* For detailed information on using the SAS program, see your SAS manual.

Purpose of the SAS interface

The SAS interface extracts resource, degradation, and profile data from the historical datastores and the Profile datastore and passes it to SAS for statistical reduction and graphic presentation.

This data can then be formatted into a variety of reports using a SAS program or added to an existing SAS performance database to enhance your current performance analysis system.

SAS interface requirements

In order to use the SAS interface, you need:

- SAS version 6.07 or 6.08
- a SAS-supported graphical device (or you will have to run the non-graphic versions of the report programs)
- a historical datastore with enough data to generate the graphs

In this section

In this section we cover the following topics.

- “Getting Started” on page 230
- “Understanding the Starter Kit” on page 232
- “Exporting Data to SAS” on page 235
- “Running the Graphics Replay Procedure” on page 238
- “Developing Your Own SAS Reports” on page 240

Getting Started

To use SAS, you must know how to supply parameters to the SAS programs, run the programs, and display the results.

SAS program parameters

Each SAS program in the *thilev*.TKANSAM dataset produces a separate report. Before running the program of your choice, you must modify certain parameters in the program by setting user overrides and providing data selection commands in the data extraction
procedure identified by the &USERPROC variable. &USERPROC resolves correctly, based on the version of SAS that is running at your site.

**Running SAS programs**
You can run SAS programs online using the sample CLIST, or in batch mode using the sample batch job.

**Displaying results**
The sample CLIST and batch job allocate and free a temporary SAS file to hold the report output of the SAS program. The report is deleted after the program is run.

If you wish to save the report output for later use, you must allocate a permanent SAS graphics output file (also known as the graphics replay file) and define it to the CLIST or batch job. You can then use the graphics replay procedure to re-display the contents of the file online or print out the graphs to another device, such as a color plotter.

**Sample jobstream allocating a graphics replay file**
The sample jobstream for allocating a graphics replay file is provided in rhilev.RKANSAM(KEPGALLC) and is shown below.

```
// ... JOB CARD ...
/*
/* THIS JOB ALLOCATES THE 'GRAFBASE' FILE USED BY KEPPMTSO and KEPPMJCL.
/* TO RUN THIS JOB:
/* 1. ADD A JOB CARD
/* 2. CHANGE THE 'SPACE' PARAMETER TO REFLECT YOUR SPACE REQMTNS
/* 3. CHANGE 'VVVVVV' TO THE VOLSER YOU WISH TO ALLOC ON
/* 4. CHANGE 'ZZZZZZZZ' TO THE HIGH LEVEL QUALIFIER OF THE DATASET
/* TO BE USED AS GRAFBASE
/*
ALLOC EXEC PGM=IEFBR14
//KEPPOUTG DD SPACE=(UUU,(P,S)), UNIT=SYSDA,
//         VOL=SER=VVVVVV,
//         DISP=(,CATLG,DELETE),
//         DSN=ZZZZZZZZ.PRO.GRAFBASE,
//         DCB=(DSORG=DA,RECFM=U,LRECL=6156,BLKSIZ=6160)
```

**Allocating a graphics replay file**
Perform the following steps to complete the procedure for allocating a graphics replay file.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Follow the comments provided in the sample jobstream.</td>
</tr>
<tr>
<td></td>
<td><strong>Recommendations:</strong></td>
</tr>
<tr>
<td></td>
<td>▪ Specify a space parameter of 5 cylinders to start</td>
</tr>
<tr>
<td></td>
<td>▪ Specify on the ZZZZZZZZ parameter your TSO user ID or another high-level qualifier that you will later define to the CLIST or batch job.</td>
</tr>
</tbody>
</table>
Once you have allocated a graphics replay file, refer to it with the GRAPH parameter of the sample report.

Each time a SAS report is generated with the OMEGAMON II starter kit, the graphics replay file is overwritten with the output of the new program. Therefore, if you want to save the graphic output, you must locate still another file and copy the current contents of the graphics replay file into it.

**Understanding the Starter Kit**

OMEGAMON II provides you with a starter kit to help you produce SAS reports. Each sample report included in the starter kit provides you with a picture of how various workloads or system resources are being utilized and which ones are constraining system performance.

**Types of SAS reports**

The model SAS reports in *thilev.TKANSAM* are grouped into three categories, as shown in the following table:

<table>
<thead>
<tr>
<th>Report Category</th>
<th>Member Name Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>KEPR</td>
</tr>
<tr>
<td>Performance Group</td>
<td>KEPP</td>
</tr>
<tr>
<td>Batch Job</td>
<td>KEPB</td>
</tr>
</tbody>
</table>

Each report has a graphic and/or character graphic (SYSPUT) format. The member names in *thilev.TKANSAM* for the character graphic version have the usual KEP prefix followed by L (with some exceptions as noted). Example:

KEPRLCBH

The tables on the next few pages describe each SAS report contained in the starter kit. Reports are grouped by type and listed with their member names in *thilev.TKANSAM*. The first member name is used to produce the report in graphic form. The second member name is used for character graphic output.

**SAS resource reports**

<table>
<thead>
<tr>
<th>SAS Report Name and PDS Member Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Channel Utilization Report</strong></td>
<td></td>
</tr>
<tr>
<td><em>thilev.TKANSAM</em>(KEPRCHBH) or (KEPRLCBH)</td>
<td>Plots the average percent utilization per hour for selected Channel Path IDs over a specified time period. By specifying all channels, you produce a separate graph for each channel.</td>
</tr>
<tr>
<td>SAS Report Name and PDS Member Names</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Channel Peak Utilization Report</strong>&lt;br&gt;thilev.TKANSAM(KEPRCHBP) or (KEPRLCBP)</td>
<td>Plots the peak busy hour of the day for selected Channel Path IDs for each day in the specified time period. Each bar on the SAS chart represents the peak hour of a particular day, its height representing the percent utilization during that hour. By specifying all channels, you produce a separate graph for each channel.</td>
</tr>
<tr>
<td><strong>Demand Paging Activity Report</strong>&lt;br&gt;thilev.TKANSAM(KEPRPAGP) or (KEPRLPAP)</td>
<td>Shows the average demand paging activity for each hour of the day. Each point on the two-dimensional SAS graph represents the average demand paging rate for that hour. Separate plots are shown for page-in and page-out rates.</td>
</tr>
<tr>
<td><strong>Swap Paging Activity Report</strong>&lt;br&gt;thilev.TKANSAM(KEPRPAGS) or (KEPRLPAS)</td>
<td>Shows average swap paging activity for each hour of the day. Each point on the two-dimensional SAS graph represents the average swap paging rate for that hour. Separate plots are shown for page-in and page-out rates.</td>
</tr>
<tr>
<td><strong>CPU Utilization by Performance Group Report</strong>&lt;br&gt;thilev.TKANSAM(KEPRPGNC) No character graphic version.</td>
<td>Shows the average hourly CPU utilization over the day broken down by performance groups. You can combine performance groups into summary groups to simplify graphic output.</td>
</tr>
<tr>
<td><strong>Real Storage Utilization by Performance Group Report</strong>&lt;br&gt;thilev.TKANSAM(KEPRPGNS) No character graphic version.</td>
<td>Shows the average hourly real storage used over the day by each performance group. You can combine performance groups into summary groups to simplify graphic output. A horizontal line is automatically plotted to indicate the maximum storage available for private working sets.</td>
</tr>
</tbody>
</table>
| **DASD Utilization Report**<br>thilev.TKANSAM(KEPRDASD) Character graphic output to printer. | Lists DASD devices in decreasing order of sustained peak utilization, which is determined as follows:  
- Data is extracted for the DASD devices during the time period you specify in the EXTRACT command, combined at 1 hour intervals.  
- For each device, an average hourly busy is calculated for each hour of the day over the days in the time period.  
- The program selects the averaged hour that had the highest percent busy value for each device. This is the *sustained peak utilization*.  
- The program sorts the devices in descending order of percent busy values. |
SAS performance group reports

<table>
<thead>
<tr>
<th>SAS Report Name and PDS Member Names</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Performance Group Response Time Degradation Report**  
  *thileu.TKANSAM(KEPPPGNP)* or *(KEPPLPGP)*  
  Shows the main wait reasons that contributed to the response time degradation for a given performance group. Produces two pie charts:  
  - The first chart represents the total response time as a pie, with each significant summary wait category constituting a piece of that pie.  
  - The second chart breaks down the largest summary wait category from the first chart into detailed wait reasons. |
| **Performance Group Trending and Forecasting Report**  
  *thileu.TKANSAM(KEPPPGNT)* or *(KEPPLPGT)*  
  Character graphic version generates only the first bar chart.  
  Examines the major wait reasons that degrade response time for a given performance group. Generates two graphs:  
  - a vertical bar chart that shows how the average response time for each day was broken down by summary wait categories  
  - a smooth trend of the response time over the selected time period, with different colored bands showing how each summary wait category contributed to the overall response time  
  SAS ETS software creates a third graph, which forecasts the performance group’s behavior for the next few days. |

SAS batch job reports

<table>
<thead>
<tr>
<th>SAS Report Name and PDS Member Names</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Batch Job Comparison Report**  
  *thileu.TKANSAM(KEPBJOBC)* or *(KEPBLJBC)*  
  Compares a profile of the last run of a batch job with a profile of an average run of that job over a specified time period. Each profile consists of a breakdown of the main summary wait categories that contributed to the run time of the job. The output of the report is a bar graph, and the bands in the bars represent summary wait categories. This program uses only information from the EPILOG datastore. |
Producing SAS Graphical Reports

Exporting Data to SAS

Once you know how SAS generates reports and understand the function of each report, you are ready to produce reports of your own.

Modeling reports

The easiest way to produce SAS reports is to model them after sample reports included in the SAS starter kit. You choose a report, modify the user overrides, and provide the required keywords.

Overview of producing reports

<table>
<thead>
<tr>
<th>SAS Report Name and PDS Member Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Batch Job Trending and Forecasting Report</strong>&lt;br&gt;thilev.TKANSAM(KEPBJOBT) or (KEPBLJBT)&lt;br&gt;Character graphic version generates only the first bar chart.</td>
<td>Examines the major summary wait categories that affect the run time for a given batch job. Generates two graphs:&lt;br&gt; - a vertical bar chart that shows how the average run time for each run of the job was broken down by summary wait categories&lt;br&gt; - a smooth trend of the run time over the selected time period, with different colored bands in the graph showing how each summary wait category contributed to overall run time&lt;br&gt;SAS ETS software creates a third graph, which forecasts the job for its next few executions.</td>
</tr>
<tr>
<td><strong>Batch Program Resource Consumption Report</strong>&lt;br&gt;thilev.TKANSAM(KEPBPGML)&lt;br&gt;Character graphic version to printer.</td>
<td>Describes the resources consumed by your batch programs. The report is produced in three parts:&lt;br&gt; - Part 1 lists batch programs that ran during the specified time period by frequency of execution. Programs are listed in decreasing order, with the most frequently run program listed first.&lt;br&gt; - Part 2 lists programs in decreasing order by CPU consumption. The program that used the most CPU time is listed first.&lt;br&gt; - Part 3 lists programs in decreasing order by active I/O time. The program that used the most active I/O time is listed first.</td>
</tr>
</tbody>
</table>
You produce reports from the starter kit with the following procedure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Locate the member in <code>thilev.TKANSAM</code> that contains the SAS program for the report you wish to produce. Report descriptions and members are found under “Understanding the Starter Kit” on page 232.</td>
</tr>
<tr>
<td>2</td>
<td>Copy this member into the RKANSAM dataset with a unique filename so that the original member remains unchanged.</td>
</tr>
<tr>
<td>3</td>
<td>Using an editor, modify the user overrides and data extraction procedure &amp;USERPROC contained in the copy as directed in the comments provided in the member.</td>
</tr>
<tr>
<td>4</td>
<td>Run the SAS program, including in the CLIST or JCL the <code>PROG(cccccc)</code> parameter, where <code>ccccc</code> is the name of the copy.</td>
</tr>
<tr>
<td>5</td>
<td>Run the graphics replay procedure (optional).</td>
</tr>
</tbody>
</table>

**User overrides**

The complete list of user overrides is described below. The overrides used by individual reports are identified in the `thilev.TKANSAM` member for that report.

- **BATNAME**  
  The name of the batch job.

- **CHANNEL**  
  Specifies the channel number you would like to report on for the channel utilization and I/O rate reports. To produce graphs for all channels in your system, enter 999.

- **DDNAME**  
  Defines the ddname of the output SAS data library specified in the CLIST or batch job that runs the report.

- **DEVICE**  
  Defines the name of the output graphic device driver. For example, if you are directing the output to a 3279 terminal, enter IBM32793.

- **FORECAST**  
  If you have SAS ETS software installed, you can run additional forecasting sections for the KEPBJOBT and KEPPPNT reports. To do so, set this override to YES. If you do not have SAS ETS, you must leave this set to NO.

- **KEPCOLOR**  
  Sets the color formatting for the output device. If the output device is a color graphics terminal, enter CRT. If you are using another output device to produce hardcopy, enter HARD. (This override is actually a macro call. You can supply your own version of the KEPCOLOR macro to match your hardware configuration.)

- **LEAD**  
  If you are running the forecasting section of the KEPBJOBT or KEPPPNT report, LEAD allows you to set the number of job runs to be forecasted. Generally, this should not exceed 10% of the number of runs in the report.
Identifying the user overrides to modify

Each of the sample programs in the starter kit contains a set of user overrides that control SAS formatting, report titling, and data selection.

A sample set of user overrides is shown in the following figure.

<table>
<thead>
<tr>
<th>USER OVERRIDES - CHANGE JJJJJJJJJJ TO DESIRED JOBNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>%LET BATNAME=JJJJJJJJJJ; /* SELECT THE JOB TO BE CHARTED */</td>
</tr>
<tr>
<td>%LET DEVICE=IBM32793;   /* SELECT GRAPHICS DEVICE DRIVER NAME */</td>
</tr>
<tr>
<td>%LET TERMINAL=TERMINAL; /* TERMINAL=SHOW GRAPHS IMMEDIATELY */</td>
</tr>
<tr>
<td>/* NOTERMINAL=DON'T SHOW GRAPHS NOW */</td>
</tr>
<tr>
<td>%LET DDNAME=EPPRO;      /* DDNAME OF &amp;USERPROC OUTPUT */</td>
</tr>
<tr>
<td>%KEPCOLOR (CRT)        /* HARD=HARD COPY, CRT=CRT SCREEN KEPCOLOR*/</td>
</tr>
</tbody>
</table>

The comments in the member tell you how to modify these overrides for this report.

Modifying the data extraction procedure
The data extraction procedure represented by &USERPROC is in every member of thileu.TKANSAM. A sample procedure is shown below.

As you see, the data extraction procedure represented by &USERPROC consists of four components:

1. The procedure &USERPROC statement, which includes the ddname parameter. This ddname is set to the SAS output file name defined in the user overrides. (&DDNAME does this automatically.)

2. The PARMCARDS line, which signals the beginning of the input cards for the procedure. This line must end with a semicolon.

3. The input cards for the procedure, which are one or more EXTRACT (EXT) or COMPEXT (CMX) commands that select the data from the EPILOG datastore and convert it into a format that is compatible with SAS processing.

   EPILOG commands cannot be written on the same line as the PARMCARDS statement, nor can they be terminated with a semicolon.


4. The terminating semicolon for the input cards must be written on the line following the last input card.

**Running the program**

There are two ways to run the program without keeping the report output.

<table>
<thead>
<tr>
<th>IF you want to run the program...</th>
<th>THEN edit...</th>
</tr>
</thead>
<tbody>
<tr>
<td>interactively</td>
<td>the sample CLIST provided in rhilev.RKANCLI(KPMTSO6).</td>
</tr>
<tr>
<td>in batch mode</td>
<td>the sample JCL provided in rhilev.RKANSAM(KPJMCL6).</td>
</tr>
</tbody>
</table>

**Running the Graphics Replay Procedure**

The graphics replay procedure re-displays SAS report output saved in the graphics replay file by directing the contents of the file to a terminal or to another device, such as a color plotter.

**Before you begin**

The graphics replay procedure will only work if you have already allocated a graphics replay file and run the CLIST or batch job to populate the output file. For complete information on allocating a graphics replay file, see “Getting Started” on page 230.
The graphics replay program

The SAS program to run the graphics replay procedure is contained in `thieu.TKANSAM(KEPGREPL)` and is shown below.

```sas
COPYRIGHT (C) 1985, AN UNPUBLISHED WORK BY CANDLE CORPORATION. ALL
RIGHTS RESERVED. THIS PROGRAM IS THE PROPERTY OF CANDLE CORPORATION
AND CONTAINS PROPRIETARY CONFIDENTIAL INFORMATION AND TRADE
SECRETS. IT IS PROVIDED ONLY FOR INTERNAL USE UNDER LICENSE FROM
CANDLE CORPORATION. IT MAY NOT BE USED COPIED OR DISTRIBUTED EXCEPT
AS AUTHORIZED UNDER SUCH LICENSE.

%INC RKANSAM(KEPPROVE) / SOURCE2 ;

+-------------------------------------------------------------------+
| COPY REPLAY STORED GRAPHS                                         |
+-------------------------------------------------------------------+
| THIS PROCEDURE MAY BE USED TO REPLAY GRAPHIC CHARTS FOR EITHER     |
| A CRT DEVICE OR PLOTTER. FOR BEST RESULTS THE REPLAY DEVICE AND   |
| THE DEVICE SPECIFIED IN THE GRAPH CREATION SHOULD BE THE SAME.    |
+-------------------------------------------------------------------+

NOTE : SAS MUST BE EXECUTED USING THE MACRO OPTION.

THIS PROGRAM IS DESIGNED TO WORK WITH SAS VERSION 5.08 AS WELL AS SAS VERSION 82.4

+-------------------------------------------------------------------+
|                     MACROS                                        |
+-------------------------------------------------------------------+
%INCLUDE RKANSAM(KEPVERSII) / SOURCE2 ;
%INCLUDE RKANSAM(KEPMREPL) / SOURCE2 ;

+-------------------------------------------------------------------+
| USER OVERRIDES                                                   |
+-------------------------------------------------------------------+
%LET DEVICE=IBM32793; /* SPECIFY THE REPLAY DEVICE DRIVER */
%LET TERMINAL=TERMINAL; /* TERMINAL=SHOW GRAPHS IMMEDIATELY */
%LET DDNAME=GRAPH ; /* NOTETERMINAL=DON'T SHOW GRAPHS NOW */
%LET DSNAME=JOBDATA ; /* DNAME OF STORED GRAPHIC DISPLAY FILE */
%LET DSNAME=JOBDATA ; /* SAS DATA SET NAME OF GRAPH FILE */

+-------------------------------------------------------------------+
| KEPGREPL                                                        |
+-------------------------------------------------------------------+
%KEPVERSII /* DETERMINE SAS VERSION BEING EXECUTED */
RUN;

GOPTIONS DEVICE=&DEVICE &TERMINAL ;
OPTIONS &TEXT82 DQUOTE ;

%KEPMREPL /* RUN THE CORRECT KEPGREPL FOR SAS VERSION */
ENDSAS;

Graphics replay program user overrides
```
The graphics replay program contains four user overrides.

**DEVICE** The output device identifier. This should correspond to the identifier entered as the DEVICE override in the SAS program. If different identifiers are entered, incompatibilities may cause such problems as visual distortion or abnormal session termination.

**TERMINAL** If a terminal was specified as the output device identifier, enter TERMINAL in this override. If another identifier was specified, enter NOTERMINAL as the override.

**DDNAME** The ddname of the graphics replay program, as defined by the ddname of the JCL used to run the procedure. If you are running reports from the starter kit, you should leave the DDNAME override set to GRAPH.

**DSNAME** The second level qualifier of the dataset name used during graph generation. In the starter kit programs, this is set to JOBDATA.

### Running the graphics replay procedure

Perform the following steps to run the graphics replay procedure:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Copy <em>thilev.TKANSAM(KEPGREPL)</em> into another member of the same dataset so that the original remains unchanged.</td>
</tr>
<tr>
<td>2</td>
<td>Modify the user overrides according to your needs.</td>
</tr>
<tr>
<td>3</td>
<td>Replay the output in one of the following ways:</td>
</tr>
<tr>
<td></td>
<td>- To replay the graphs on a terminal, run the sample CLIST with the name of the member into which you copied KEGREPL as the <em>cccccc</em> in the PROG(cccccc). For example, if you copied KEGREPL into a member named MYREPLAY, enter:</td>
</tr>
<tr>
<td></td>
<td>%KPMTSO6 PROG(MYREPLAY)</td>
</tr>
<tr>
<td></td>
<td><strong>Result:</strong> A menu containing each graph generated by the report is displayed. Select the graphs you want to see by placing an S next to the graph name on the menu.</td>
</tr>
<tr>
<td></td>
<td>- To replay the graphs to a hardcopy device, run the sample batch job with the name of your copy of KEGREPL as the PROG parameter.</td>
</tr>
</tbody>
</table>

### Developing Your Own SAS Reports

Now that you know how to produce SAS reports, it is time to develop two examples.

#### Developing a resource report

Suppose you want to produce a Channel Utilization Report and direct the output to your terminal:
Locate the PDS member name *thileu.TKANSAM (KEPRCHBH)* for the report in “Understanding the Starter Kit” on page 232 and copy its contents to another member, which you call MYCOPY.

Edit MYCOPY and modify the six user overrides and three EXTRACT keywords contained in the data extraction procedure &USERPROC.

- Run the report by executing the CLIST, specifying PROG(cccccc) where cccccc is MYCOPY. Enter:

  ```
  %KPMTSO6 PROG(MYCOPY)
  ```

  The first graph is displayed. If you specified CHANNEL=999 in the user overrides, a graph is produced for each channel in your system. Press F3 or F15 to display the next graph.

**Developing a Performance Group Response Time Degradation Report**

Now you want to produce a Performance Group Response Time Degradation Report. This time you want to run the report using the batch job and then use the graphics replay procedure under TSO to display it on the terminal.

The initial steps are the same as before except that the original member is *thileu.TKANSAM(KEPPPGNP)*. The user overrides and data extraction procedure
&USERPROC are shown below. This time you must modify seven user overrides and four EXTRACT keywords.

Submit the following job:

```
//JOB...
// EXEC KPMJCL6, PROG=MYCOPY
```

When the job finishes, copy thileu.TKANSAM(KEPGREPL) into another member of the same dataset, edit the user overrides, and execute the CLIST with your member name for KEPGREPL as the `cccccc` of PROG(`cccccc`).

**SAS macros**

Nine macros were used to write the programs for the starter kit reports, and you may want to use or modify these macros if you write your own programs.

The following table provides a definition of each macro.

<table>
<thead>
<tr>
<th>Macro</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEPBTCHM</td>
<td>Used to condense and simplify data in the batch job trending report. The macro groups several related major wait reasons into summary wait categories.</td>
</tr>
<tr>
<td>KEPCOLLS, KEPCOLLCC</td>
<td>When a starter-kit program generates more than one graph with SAS, KEPCOLLS starts the process of collecting them from a temporary dataset by writing the first graph to a permanent dataset (GRAPH.JOBDATA). KEPCOLLCC continues the process by writing the subsequent graphs to the same permanent dataset.</td>
</tr>
</tbody>
</table>
### Macro | Definition
--- | ---
**KEPCOLOR** | Sets eight global macro variables to define the color scheme on the output device. The version that is distributed with the product has an input variable that can be set to CRT or HARD. CRT sets the color scheme for an IBM 3279 terminal. HARD sets the color scheme for a color plotter. You should modify the color settings for the HARD option to correspond with your hardcopy output device, if necessary.

**KEPDELGR** | When a starter-kit program generates a new graph(s) with SAS, this macro deletes the old version(s) from the graphics replay file.

**KEPMREPL** | Conditionally executes the KEPGREPL procedure. It is used only in the starter-kit program KEPGREPL.

**KEPPGNMA** | Used to condense and simplify data in the performance group trending and response time degradation reports. The macro groups several related wait reasons into a summary group that represents a major wait reason.

**KEPSUMMG** | Simplifies defining a summary report group. In the starter-kit programs KEPRPGNC and KEPRPGNS, you can use this option to combine the data for several performance groups and define them as a single reporting group, such as TSO, CICS, or BATCH. You must first modify this macro to correspond with performance group definitions at your installation.

**KEPVERSI** | Uses the SAS automatic variable &SYSVER to determine which version of SAS is being executed. It then modifies release-dependent macro variables accordingly.
Producing SAS Graphical Reports
Section 2
WLM-Based Performance Monitoring
Introduction

The OMEGAMON II System Status panel is the starting point for monitoring your MVS system. To resolve performance problems on your system you need to understand how to interpret the status indicators, navigate through the product in response to status alerts, and track a problem to its source.

The examples presented in this chapter are designed to help you learn how to monitor your system when it is running in goal mode.

Chapter Contents

Interpreting Status Lights ................................................................. 248
Monitoring Workloads, Resources, and Alerts ........................................ 254
Monitoring Performance across Systems ............................................. 307
Interpreting Status Lights

Introduction
The System Status panel compares the current performance of monitored objects to their threshold values. The thresholds are set against the performance goals at your site, as defined by your site’s MVS service-level administrator. If a problem exists, a light or symbol will appear in a status indicator area dynamically alerting you to the problem.

In this section
In this section we cover the following topics:
- “Status Lights on the System Status Panel” on page 248.
- “General Guidelines for Handling Status Alerts” on page 250.
- “General Guidelines for Monitoring a Healthy System” on page 252.

Status Lights on the System Status Panel

Introduction
The panel body of the System Status panel is organized into three status indicator areas:
- Workload status
- Resource status
- Operator Alerts
Within the status indicator areas, colors or symbols are used to signal status alerts. In the screen below, we use symbols ($$$$$, *****, and -----) to show the status of the monitored objects in the panel body.

Colored Lights and Associated Conditions
On a color terminal, items on the panel body can be any one of four colors, and each color signals a different system condition.

The four colors and the conditions they indicate are shown below.

<table>
<thead>
<tr>
<th>Color</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>Normal</td>
</tr>
<tr>
<td>Yellow</td>
<td>Warning (there is a possible system problem)</td>
</tr>
<tr>
<td>Red</td>
<td>Critical condition (there is a definite system problem)</td>
</tr>
<tr>
<td>Turquoise (blue on some monitors)</td>
<td>Monitoring of object is idle (data is not available because RMF is inactive) or disabled (using the L action code)</td>
</tr>
</tbody>
</table>

Symbols and associated conditions
If your terminal does not support extended attributes or is not a color terminal, symbols instead of colors are used as status indicators.
Interpreting Status Lights

The symbols and conditions indicated are shown below.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyphen (-)</td>
<td>Normal condition</td>
</tr>
<tr>
<td>Asterisk (*)</td>
<td>Warning condition</td>
</tr>
<tr>
<td>Dollar sign ($)</td>
<td>Critical condition</td>
</tr>
<tr>
<td>Blanks</td>
<td>Monitoring of object is idle or disabled</td>
</tr>
</tbody>
</table>

*Note: The symbols above are the default settings.*

You can change the default settings of the symbols if you wish. To change the default settings:

1. Select the Options pull-down menu.
2. Select Controls from the Options pull-down.
3. Select Session defaults and change the appropriate fields.

General Guidelines for Handling Status Alerts

Introduction

On the following pages we present two tables for handling status alerts:

- for workloads or resources (left side of the System Status panel)
- for operators (right side of the System Status panel)

In both tables we present actions you might take to meet a variety of objectives. As a reminder for new users, start with s for Show Details as a first step and then proceed through the other steps.

As you grow more familiar with the product, you will learn where Bottlenecks or Analyze Problems are productive alternatives as a first step.

Responding to status alerts for workloads or resources

You can investigate the status of any object on the panel at any time, regardless of the color of the light. But, when you see a red or yellow status light for workload or resource status on the left side of the System Status panel, the following suggestions for action may be useful to you in your investigation.

<table>
<thead>
<tr>
<th>IF the objective is to see...</th>
<th>THEN...</th>
<th>TO see this result...</th>
</tr>
</thead>
<tbody>
<tr>
<td>details about an object</td>
<td>type s for Show Details on the input line before the item</td>
<td>a panel of details that allows you to begin to isolate the problem area.</td>
</tr>
<tr>
<td>a list of conditions that exceed thresholds</td>
<td>type a for Analyze Problems</td>
<td>a comparison between actual and expected performance for a specific indicator.</td>
</tr>
</tbody>
</table>
### Table: Interpreting Status Lights

<table>
<thead>
<tr>
<th>IF the objective is to see...</th>
<th>THEN...</th>
<th>TO see this result...</th>
</tr>
</thead>
<tbody>
<tr>
<td>the trends in performance for an object in the past</td>
<td>type t for Historical Trends for the past performance</td>
<td>how a workload or resource has performed over a number of intervals</td>
</tr>
<tr>
<td>how an object has performed within a specified time period</td>
<td>type h for Historical Details for past performance</td>
<td>detailed information for a selected time period combined into a single averaged result</td>
</tr>
<tr>
<td>whether a bottleneck exists</td>
<td>type s for Show Details to take you one level below the System Status panel. Then type b for Bottlenecks on the input line before the job or service class. <em>Bottleneck analysis is appropriate for objects such as address spaces, for example.</em></td>
<td>the Bottlenecks panel, enabling you to identify conditions affecting workload performance.</td>
</tr>
<tr>
<td>additional related information</td>
<td>type g for GoTo on the input line of the action bar</td>
<td>the GoTo pull-down menu, enabling you to choose related information that might help you diagnose the problem.</td>
</tr>
<tr>
<td>closely related information about the same object</td>
<td>place the cursor on the appropriate pushbutton at the bottom of a panel and press Enter</td>
<td>panels closely related in subject and level of detail to the current panel.</td>
</tr>
</tbody>
</table>

### Responding to status alerts for operators

You can investigate the status of any object on the panel at any time no matter what color the status light is.
But, when you see a red or yellow status light for operator alerts on the right side of the System Status panel, the following suggestions for action may be useful to you in your investigation.

<table>
<thead>
<tr>
<th>IF the objective is to see...</th>
<th>THEN...</th>
<th>TO see this result...</th>
</tr>
</thead>
<tbody>
<tr>
<td>details for an object</td>
<td>type <code>s</code> for Show details on the input line before the object</td>
<td>for all indicators except HSM, SMF, and Enqueue, a message indicating the problem. For HSM, SMF, and Enqueue, a detail panel is displayed.</td>
</tr>
<tr>
<td>the System Console panel</td>
<td>press Enter on the object input line and then press Enter again from the pop-up window (or press F6)</td>
<td>the System Console panel, enabling you to enter MVS or other commands.</td>
</tr>
<tr>
<td>additional related information</td>
<td>type <code>g</code> for GoTo on the input line of the action bar</td>
<td>the GoTo pull-down menu, enabling you to choose related information that might help you diagnose the problem.</td>
</tr>
<tr>
<td>closely related information about the same object</td>
<td>place the cursor on the appropriate pushbutton available at the bottom of some panels (not the System Status panel) and press Enter</td>
<td>panels closely related in subject and level of detail to the current panel.</td>
</tr>
</tbody>
</table>

General Guidelines for Monitoring a Healthy System

Introduction

There are times when you want to investigate some aspect of a particular object’s performance, though at the time no red or yellow status light is showing for the object.

This section gives some guidelines for investigating objects which show only green (for normal) or turquoise (for idle) on the System Status panel.

The same navigating concepts apply regardless of the color of the status light. You navigate around the system the same way whether the status lights are green for normal, yellow for warning, or red for critical.

Investigating a healthy system
The table that follows presents some general guidelines for exploring an object with OMEGAMON II though its status light is green.

<table>
<thead>
<tr>
<th>IF you want to...</th>
<th>THEN...</th>
</tr>
</thead>
<tbody>
<tr>
<td>see a panel of details about an object</td>
<td>type s for Show Details on the input field before the object.</td>
</tr>
<tr>
<td>see the trends in performance for an object over a period of time</td>
<td>type t for Historical Trends on the input field before the object.</td>
</tr>
<tr>
<td>see detailed information on how an object has performed over a selected time period, combined into a single averaged result</td>
<td>type h for Historical Details on the input field before the object.</td>
</tr>
<tr>
<td>see additional related information on the GoTo pull-down menu</td>
<td>type g for GoTo on the input line of the action bar.</td>
</tr>
<tr>
<td>see closely-related information about an object</td>
<td>place the cursor on the appropriate pushbutton at the bottom of a panel, where available, and press Enter. Accessible pushbutton panels are enclosed in &lt; &gt;.</td>
</tr>
</tbody>
</table>
Monitoring Workloads, Resources, and Alerts

Introduction

In this section we present examples of how you might use OMEGAMON II to monitor your system’s performance. We have included at least two examples from each of the three status indicator areas.

- Workload status - Six examples: current and recurring TSO response time problems, excessive elapsed time, I/O wait, and batch and STC problems
- Resource status - Five examples: CSA problem, cache statistics (DASD-related), replacing link list datasets, and channel I/O and CPU utilization problems
- Operator Alerts - Two examples: key task and enqueue conflict

The procedures used in the examples do not vary greatly from the procedures you would use for any of the other objects found on the System Status panel. We encourage you to go through each example to learn how to monitor performance in each status indicator area.

In this section

In this section we cover the following topics.

- “Example: Investigating a Current Response Time Problem” on page 254
- “Example: Investigating a Recurring Response Time Problem” on page 257
- “Example: Investigating Excessive Elapsed Time” on page 261
- “Example: Resolving an I/O Wait Problem” on page 265
- “Example: Resolving a Batch Problem” on page 272
- “Example: Resolving a Started Task Problem” on page 274
- “Example: Investigating a CSA Problem” on page 280
- “Example: Looking Up Cache Statistics” on page 283
- “Example: Replacing Link List Datasets” on page 289
- “Example: Resolving a Channel I/O Problem” on page 293
- “Example: Resolving a CPU Utilization Problem” on page 299
- “Example: Investigating a Key Task Alert” on page 301
- “Example: Responding to an Enqueue Conflict” on page 304

Example: Investigating a Current Response Time Problem

Introduction

If your thresholds are set correctly, you should see a response time problem on the System Status panel before anyone calls the data center to complain.

Example of poor response time
Suppose you are monitoring your system from the System Status panel, and you see the TSO: RTA™ status light turn red, as pictured in the following figure.

To investigate the cause of the red light, you issue the **Show details** action for this indicator by moving the cursor to the TSO: RTA input field and pressing Enter. Issuing the **Show**
details action for the TSO: RTA indicator on the System Status panel leads you to the TSO Response Time Groups panel pictured below.

To choose a TSO group for further investigation, consider the following factors:

- Are any columns highlighted yellow or red?
- Are there unusually high or low values in the host response or network response columns?

In this example, the San Francisco group meets these criteria, with a network response time of 22.63 seconds. Before calling your Network Support group, you issue the :xph.Show details:exph. action for San Francisco to gain more information regarding the network delay.
Issuing the **Show details** action on the TSO Response Time Groups panel displays the TSO Response Time Group Users panel pictured below.

The details confirm that the red light on the System Status panel is being caused by a network problem. You phone Network Support while the details of the San Francisco activity remain on your OMEGAMON II display.

**Example: Investigating a Recurring Response Time Problem**

**Introduction**

The following example shows how to use a response-time distribution to investigate a recurring response time problem.

**Operations discovers the problem**

Suppose the Operations staff phones you to report the following recurring problem:

When the Per1 TSO light turns red and Operations displays realtime details, a particular TSO service class, TSODEV, is frequently performing significantly over its response time goal. The users in this service class, however, have *not* complained of slow response time.

**Normal investigation provides no answers**

To attempt to investigate this phenomenon, Operations has repeatedly displayed the Service Class Period Workflow Analysis panel, but it shows no resource contention to justify the red light. However, there is a high Idle time for TSODEV, and since this is unusual, they request more details.

The resulting Bottlenecks panel for TSODEV reveals a high ECB Wait (w/ STIMER).

**Checking for an unexpected red light**

Because your users are satisfied, it appears that your system is displaying a workload warning or critical light that is unexpected or misleading. So as a first step, you bring up the Set...
Service Class Thresholds panel. On this panel, if the Goal Type is specified as Velocity, then an updatable field, **with CPU percent busy**, is presented to you. In that field, you can specify a percentage to identify the minimum percentage of CPU that must be consumed before OMEGAMON II for MVS displays the warning or critical light.

**A new line of investigation**

If the **with CPU percent busy** value does not clarify the situation, you can proceed further. Given the satisfied users, you suspect that the use of the STIMER service is involved in triggering the red light. Programs that include the use of a wait timer often produce what appear to be long response times.

The first time you see the Per1 TSO light turn red on the System Status panel, you request more details for that light by moving your cursor to its input field and pressing Enter. The TSO version of the Workload Manager Overview panel appears, as pictured below.

**Identifying the problem service class**

The TSO version of the Workload Manager Overview panel displays WLM performance statistics for period 1 of all active TSO service classes, and a single status indicator that represents the worst of periods 2 - 8 for each service class.

The panel contents are as Operations described. Period 1 of service class TSODEV has a performance index (P/I) of 2, indicating that its Actual average response time is twice its average response-time goal. (The term performance index is described below.)

You want to investigate the statistical make-up of TSODEV's first period Actual average response time. To do so, you enter **s** in the Actual response time input field for TSODEV.

**What is a performance index?**

A performance index, represented by P/I on OMEGAMON II panels, is a measure of how a service class period is performing compared to its WLM performance goal, regardless of the goal type or the units in which the goal is specified.
A performance index of 1.0 means that a service class period is performing at its requested goal. As the actual performance of a service class period deteriorates, its performance index increases. For example, if a service class period has an average response time goal of 2 seconds, and its current actual average response time is 6 seconds, its current performance index is 3.0.
Looking at a response time distribution

Entering $s$ in an Actual response time input field of the Workload Manager Overview panel displays the Response Time Distribution pop-up window pictured below.

The distribution of the response times for period 1 of TSODEV helps explain why the service class is frequently missing its average response time goal, yet maintaining satisfied users.

According to the display, 83% of the transactions in this service class period are receiving response times as good as or better than the performance goal, and only 2 or 3 transactions are achieving response times worse than 1.5 times goal.

At a glance, you can see that without the 2 or 3 transactions in the >525 ms column the average response time is less than the 350 ms goal. In order to achieve an Actual average response time of 700 ms, the 2 or 3 transactions must be achieving response times that are significantly beyond the scale of this graph. These few statistical anomalies are skewing the average response time of the entire service class and causing the Per1 TSO light to turn red on the System Status panel.

The red light with the satisfied users is no longer a mystery. Nevertheless, you decide to investigate the statistical anomalies; you are looking for the user with the high response times.

Investigating further

To further investigate the users who make up the statistical anomalies, you list the users in TSODEV along with their response times by entering fast path iwru in the action bar input field.

OMEGAMON II displays the TSO Users Overview panel. You notice user MARYK with a transaction that has been processing for 28 seconds. You phone MARYK and ask what kind of TSO response time she is receiving. She says she does not know because she is running a program that examines some specified jobs and updates their job status on her terminal every 30 seconds.
Resolving the problem
You have discovered why Operations found the ECB Wait (w/ STIMER) for TSODEV. The 30-second periods are included in MARYK’s response times, and these values are skewing the average response time for TSODEV. To accommodate MARYK’s program within the TSODEV service class, you decide to change the goal type of TSODEV from Average Response Time to Percentile Response Time.

Example: Investigating Excessive Elapsed Time

Introduction
This data center problem demonstrates that the biggest bottleneck is not always the cause of an elapsed time exception.

The following pages demonstrate how to use the OMEGAMON II for MVS historical trends and details panels to discover what is really interfering with the normal execution of a batch job.

Discovering the problem
You are monitoring your system from the System Status panel, and the Batch light turns yellow.

<table>
<thead>
<tr>
<th>Workload status</th>
<th>Resource status</th>
<th>Operator Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>S Batch  *****</td>
<td>_ CPU -----</td>
<td>_ Key Task -----</td>
</tr>
<tr>
<td>_ STC/APPC -----</td>
<td>_ DASD *****</td>
<td>_ WTO Buffer ----</td>
</tr>
<tr>
<td>_ TSO: RTA -----</td>
<td>_ Tape -----</td>
<td>_ WTORs -----</td>
</tr>
<tr>
<td>_ TSO Host -----</td>
<td>_ Paging -----</td>
<td>_ OLTEF -----</td>
</tr>
<tr>
<td>_ Per1 TSO -----</td>
<td>_ Storage -----</td>
<td>_ DDR Swap -----</td>
</tr>
<tr>
<td>_ Workload -----</td>
<td>_ CSA -----</td>
<td>_ Max. Tasks -----</td>
</tr>
<tr>
<td>_ Channels -----</td>
<td>_ HSM -----</td>
<td>_ Key DASD --</td>
</tr>
</tbody>
</table>

To request more information for this indicator, you enter S in the Batch input field. This leads you to the Batch Jobs Overview panel shown in the following figure.

On this panel, you see that job PIVJIT25 has an Elapsed time of 35:56 minutes and is the only job highlighted in yellow.
Now you know that the yellow Batch light on the System Status panel was caused by excessive elapsed time for job PIVJIT25. That is, the elapsed time for PIVJIT25 exceeds the elapsed time warning threshold of the job’s performance group.

To find out why the elapsed time is so high for this job, you enter S in the PIVJIT25 input field to reach the Details for a Job or Started Task panel shown in the following figure.

**Investigating excessive elapsed time**

The Status area on the left side of the panel shows that the elapsed time for PIVJIT25 continues to increase; it has reached 36:10 minutes.
The Elapsed time Profile area on the right side of the panel shows that CPU wait is the main bottleneck (31 percent) and I/O wait is the secondary bottleneck (20 percent). Your first suspicion is that CPU wait caused the excessive elapsed time for PIVJIT25.

Before yielding to the temptation to reduce CPU wait time by moving PIVJIT25 to a performance group with a higher dispatching priority, you decide to find out if a high CPU wait percentage is normal for this job.

If high CPU wait is normal for this job, then the I/O wait must be causing the excessive elapsed time, even though that percentage is lower. If high CPU wait is not normal, then it is the cause of the excessive elapsed time. The historical trends panel can provide this information.

To reach the historical trends panel for this job, you:

- Select Historical Trends for a Batch Job or TSO User from the GoTo pull-down menu.
- When prompted for a date and time period, you enter 3 to request this hour for the previous days of this week.

In this example, you only need data for the previous two or three executions of this daily batch job.
The trends panel for PIVJIT25 shows that CPU wait was also the major wait reason for the previous three executions of the job, though the elapsed times were all less than 30 minutes.

You want to compare the CPU wait percent of one of the previous executions displayed on this panel with the CPU wait percent of the current job, which was 31 on the previous panel. If the CPU wait percent of the current job is significantly higher than one of the previous executions, then CPU wait can still be the cause of the excessive elapsed time of the current job.

You enter an S in the input field of one of the previous time periods to display historical details for that execution of the job.

The Bottlenecks area on the right side of this panel shows that the CPU wait percent (33.46) of yesterday’s successful execution of PIVJIT25 was higher than the the CPU wait percent of the current execution (31). However, despite the higher CPU wait percent, the elapsed time was only 28.55 minutes for yesterday’s execution.
Clearly, the seemingly high CPU wait for the current execution is not the cause of its excessive elapsed time.

Now that you know a high CPU wait is normal for this job, you look to see if any other waits are normal. No other wait reasons appear in the Bottleneck data for yesterday’s successful execution.

The real cause of today’s problem

You have discovered that the biggest bottleneck that appeared earlier today on the Details for a Job or Started Task panel (31 percent) was a normal occurrence for this job, even during those executions when it met service-level expectations.

The I/O wait of 20 percent that appeared on the same panel, is the real cause of the excessive elapsed time.

Example: Resolving an I/O Wait Problem

Introduction

The following pages demonstrate how to use the OMEGAMON II for MVS seek analysis panel to discover what is interfering with the normal execution of a batch job.

Discovering the problem
You are monitoring your system from the System Status panel, and the Batch light turns yellow.

To request more information for this indicator, you issue the **Analyze Problems** action for Batch. To do this, position your cursor next to Batch, type **A**, and press Enter.

On this panel, you see that job PIVJIT25 has an Elapsed time of one hour fifteen minutes and is the only job highlighted in yellow.

Now you know that the yellow Batch light on the System Status panel was caused by an excessive elapsed time for job PIVJIT25. That is, the elapsed time for PIVJIT25 exceeds the elapsed time warning threshold of the job’s performance group.

To find out why the Elapsed Time is so high for this job, you enter **S** in the PIVJIT25 input field to reach the Details for a Job or Started Task panel shown in the following figure.

**Investigating an I/O wait problem**
The Elapsed Time Profile area on the right side of the panel shows that I/O wait is the main bottleneck (31 percent) and CPU wait is the secondary bottleneck (20 percent). Your first suspicion is that I/O wait caused the excessive elapsed time for PIVJIT25.

You enter S in the I/O wait execution state input field to display the Bottlenecks panel as shown in the following figure.

The Contention by Resource area on the right side of the panel shows that there are other jobs causing the delay.

You now want to investigate which other workloads are in contention for volume PRI003.
You enter S in the Disk PRI003 queued state input field. This action gives you the DASD Details for a Device panel, as shown in the following figure.

The Response Time area on the right side of the panel shows a high disconnect time of 33.3 ms which may indicate Seek delays.

To investigate possible Seek delays place the cursor on the (Seek Analysis) push button and press Enter. The DASD Seek Analysis for a Device panel appears, as shown in the following figure.

This DASD Seek Analysis for a Device panel graphically shows you:

- the head movement on volume PRI003
- the contention for the head exists between workloads PIVJIT25 and TOMTST12
Such contention creates Seek delays which help to account for the unusually high Queued I/O percentage displayed in the Bottlenecks panel.

You now want to place TOMTST12 into a performance group with an MPL of (0,0).

To do this, enter fast path :xph.iwda:exph. to access the All Domains panel where you can see which performance group possesses the required characteristics of a minimum MPL of 0 and a maximum MPL of 0. The All Domains panel is shown in the following figure.

**Note:** Save the current jobname, TOMTST12, so that you can return the job to the same performance group when you are ready to resume execution.

**Solving an I/O wait problem**
On this panel, you see that Domain 111 has a minimum MPL of 0 and a maximum MPL of 0.

You now want to check the current IPS to find out which performance group domain 111 belongs to. To do this, enter fast path **icse** to display the System Environment panel, as shown in the following figure.
On this panel, you see that the system parameter is IPSGG. Browse the IEAIPSGG member of SYS1.PARMLIB. You see that domain 111 belongs to performance group 13. Now you are ready to reset the performance group of TOMTST12.

Enter fast path iwbd to reach the Details for a Batch Job or Started Task pop-up window, as shown below.

```
KM2K05D                     System Environment  System: SYSA

| CPU model . . : 9672 | MVS level . . . : SP4.3.0 |
| Mode. . . : Partitioned | RMF level . . . . : 4.3.0 |
| Serial number : 020229 | RMF Monitor I status. : ACTIVE |
| IPL date/time : 09/09/99 18:12:40 | RMF Cycle length. . : 1000 |
| IPL volume. . : MT430G(04F0) | RMF Interval start. . : 8:29:06 |
| ESCON Status. : Enabled | RMF Interval length . : 14:53 MN |
| System parms. : IPSGG ICSGG OPTGG | LPAR Elapsed time . . : 0:34:366 |
```

```
Logical Partition Information

```

```
| Name  | Cur | # | Status| Wts|Wait| Cap | #LP | LCPD% | PCPD% | OVHD% |
|-------|-----|---|-------|----+----+-----+-----+--------+--------+--------|
| SP21  | 01  | Active | 001 | NO | NO |   1 |  0.00 |  0.00 |  0.00 |
| SP22  | 02  | Active | 349 | NO | NO |   4 | 49.40 | 32.94 |  0.56 |
| SP23  | 03  | Active | 650 | NO | NO |   6 | 55.08 | 55.08 |  0.88 |
| OVERHEAD | 00 |       |     |    |     |     |        |  88.02 |   2.06 |
| TOTAL |     |       |     |    |     |     |        |  88.02 |   3.50 |
```

```
<CPU Utilization> (System Environment)
F1=Help   F2=Keys   F3=Exit   F5=Refresh   F6=Console    F10=Action Bar
F11=Print     F15=System Status
```

Enter TOMTST12 in the Name of batch job or started task input field and press Enter. The Details for a Job or Started Task panel is displayed, as shown in the following figure.

**Note:** Save the current performance group of TOMTST12, so that you can return the job to the same performance group when you are ready to resume execution.
Enter fast path \textbf{ar} to reach the Reset Performance Group pop-up window. Enter 13 in the Performance Group Number input field.

When job PIVJIT25 has completed you can reset the saved performance group of TOMTST12 so it can resume execution.

\textbf{Example: Resolving a Batch Problem}

\textbf{Introduction}

The following pages demonstrate how to use the OMEGAMON II to investigate a problem in batch processing.

\textbf{Discovering the problem}
You are monitoring your system from the System Status panel, and you notice that the BATCH indicator is red.

To request more information for this indicator, you issue the **Show Details** action for BATCH. To do this, position your cursor next to BATCH, type **S**, and press Enter. This action displays the Batch Jobs Overview panel shown in the following figure.

### Investigating the problem

You notice that job WORKWK1 has a long WAIT time. To show the details for this job, position the cursor on the input area, and type **S**. This will display the Details for a Job or Started Task panel as shown in the following figure.
### Solving the problem

You notice that job WORKWK1 is waiting for a tape to be mounted. You can call Operations to ask them to mount the tape.

#### Example: Resolving a Started Task Problem

**Introduction**

The following pages demonstrate how to use the OMEGAMON II Started Tasks Overview panel to discover what is interfering with the normal execution of jobs.

**Discovering the problem**
You are monitoring your system from the System Status panel, and the STC/APPC status light turns red.

To request more information for this indicator, issue the show details action for STC/APPC. To do this, position the cursor next to STC/APPC, type `S`, and press Enter. This will display the Started Tasks Overview panel as shown in the following figure.

<table>
<thead>
<tr>
<th>Workload status</th>
<th>Resource status</th>
<th>Operator Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ Batch_ -----</td>
<td>_ CPU_ ------</td>
<td>_ Key Task_ -----</td>
</tr>
<tr>
<td>S STC/APPC $$$$$</td>
<td>_ DASD_ ------</td>
<td>_ WTO Buffer_ ------</td>
</tr>
<tr>
<td>_ TSO: RTA_ -----</td>
<td>_ Tape_ ------</td>
<td>_ WTORs_ ------</td>
</tr>
<tr>
<td>_ TSO Host_ -----</td>
<td>_ Paging_ ------</td>
<td>_ OLTEP_ ------</td>
</tr>
<tr>
<td>_ Per1 TSO_ -----</td>
<td>_ Storage_ ------</td>
<td>_ DDR Swap_ ------</td>
</tr>
<tr>
<td>_ Workload_ -----</td>
<td>_ CSA_ ------</td>
<td>_ Max. Tasks_ ------</td>
</tr>
<tr>
<td>_ Channels_ ------</td>
<td>_ HSM_ ------</td>
<td>_ Key DASD_ ------</td>
</tr>
</tbody>
</table>

To request more information for this indicator, issue the show details action for STC/APPC. To do this, position the cursor next to STC/APPC, type `S`, and press Enter. This will display the Started Tasks Overview panel as shown in the following figure.
Investigating the problem

You notice that job TSL2S20 is showing a high I/O rate and high CPU utilization. For more detailed information, position the cursor in the input area in front of the desired job, and type "/". Return the cursor to the actions input area at the top of the panel, and type V to activate the pull-down action menu.

Select the option that allows you to view the started tasks sorted by I/O rate. Investigate the job with the highest I/O rate to determine if DDNAME contention is causing the problem. Position the cursor in the input area of the desired job, type B, and press Enter. This will
display the Bottlenecks panel showing active or queued I/O as illustrated in the following figure.

You notice that job TSL2S20 is waiting for CPU processing and PRI041 on device number 0280. For detailed information, position the cursor in the input area in front of the device number, and type / . This will cause the actions pull-down menu to appear as shown in the following figure.
To display more detailed information, select the DDNAME activity in the action pull-down menu. Position the cursor at the input field for the action pull-down menu, and type D for DDNAME activity.

The address space dataset allocations pop-up panel will be displayed as shown in the following figure.
Solving the problem

---

**Actions**

GoTo  Index  Options  Help

---

<table>
<thead>
<tr>
<th>KM2W09D</th>
<th>Bottlenecks</th>
</tr>
</thead>
<tbody>
<tr>
<td>System: SP11</td>
<td>AUTO(60)</td>
</tr>
<tr>
<td>To analyze a different workload, enter jobname or Service Class</td>
<td></td>
</tr>
</tbody>
</table>

**Jobname:** TSL2S20  **Service Class:** TEST

---

**Impact by Service Class**

<table>
<thead>
<tr>
<th>Contention by Resource</th>
</tr>
</thead>
</table>

---

**KM2D08D**

**Address Space Dataset Allocations**

---

F1=Help    F2=Keys    F3=Exit    F5=Refresh    F6=Console    F10=Action

F11=Print    F15=System Status
To display more detailed information, position the cursor next to the desired dataset name in the pop-up panel, and type s. This will cause the Dataset Details pop-up panel to appear as shown in the following figure.

The Dataset Details pop-up panel does not indicate a clear cause for contention in this example. The problem may be caused by other LPARs sharing access to device number 0280. You have to repeat this example on one or more of the other LPARs to see if they are using device number 0280 and causing the problem. It may be necessary to move some of the datasets to a different device, or restrict access to device number 0280 to eliminate the cross LPAR contention.

Example: Investigating a CSA Problem

Introduction
The CSA status indicator is found in the second column of the System Status panel, under the heading “Resource status.” When you spot a potential problem with common storage (a yellow alert CSA status indicator), you can navigate to the Common Storage panel for more
details. From there, you can identify which area of CSA is having trouble, show details on that problem area, and see who is using how much common storage.

The following example shows how you might use OMEGAMON II to resolve a typical CSA problem.

**Example of CSA approaching the limit**

Suppose you are monitoring your system from the System Status panel when you notice that the CSA status indicator is yellow. First, you request the **Show details** action for CSA. To do this, position your cursor next to CSA and press Enter. This displays the Common Storage Utilization panel, shown below.

This panel lists four important areas of common storage. For each area, the panel provides the following information:

- amount of storage currently in use
- total size of each area, as specified at IPL in member IEASYSxx of SYS1.PARMLIB
- amount of allocated but unowned storage (storage that was not freed when an address space terminated)
- growth in use since IPL

Areas that are highlighted on this panel are reaching dangerously high levels of allocation.

Since initial program load (IPL), CSA has grown by 300K. You request the **Show details** action for CSA. To do this, position the cursor next to CSA and press Enter. OMEGAMON II displays the Active Users of CSA panel.
This panel lists the names and address space identifiers (ASIDs) of the current CSA users. It also lists the current amount of CSA that is unowned, the CSA growth rate, and the percentage of CSA in use.

As you view the Active Users of CSA panel, you notice two things:

1. No one user seems to have particularly high CSA usage.
2. The value for unowned storage is high for your site.

You decide to examine CSA unowned storage more closely. Use the GoTo pull-down or the <Unowned> pushbutton to request the CSA Unowned Storage panel. OMEGAMON II
displays the CSA Unowned Storage panel. This panel breaks down CSA unowned storage by area.

You notice that unowned storage is high because IMSTST left a lot of storage behind after it terminated.

**Taking action**

The action you take from here depends on your site. For this case study, assume that because the CSA allocation was so high and the possibility of a system failure is becoming more and more likely, you decide to free the storage that IMSTST left behind by using the OMEGAMON II CSAF immediate command.

To issue the CSAF immediate command, enter fast path **go**, then enter CSAF. For more information on the CSAF command, see the *OMEGAMON for MVS Command Language Reference Manual*.

After using the CSAF command, enter **end** on the INFO-line to return to OMEGAMON II.

**Example: Looking Up Cache Statistics**

**Introduction**

Because cache memory is used to reduce access time and therefore increase performance, OMEGAMON II provides a number of statistics regarding cache memory that help you fine tune your system’s performance.
OMEGAMON II reports cache statistics for cache controllers including models 3880-13, 3880-23, and 3990-3, 3990-6, and 2105. These statistics include cache read and write hit percentages and I/O requests indirectly related to cache.

**The panels you need**

The information presented on the following pages shows how to access the OMEGAMON II panels that allow you to control the monitoring of cache memory and display cache statistics.

**Setting DASD exception thresholds**

You can set thresholds relating to cache on the Set DASD Exception Thresholds pop-up, which includes four thresholds related to cache.

To access the Set DASD Exception Thresholds pop-up, follow this two-step procedure:

- From any OMEGAMON II panel, enter fast path `otd` to reach the Specify DASD Groups pop-up.
From this pop-up, select the DASD threshold group you want to change.

```
<table>
<thead>
<tr>
<th>Volume</th>
<th>TEMP03 +</th>
<th>Subsyste</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device #</td>
<td>0161 +</td>
<td>Cache Hi</td>
</tr>
</tbody>
</table>

To add a DASD threshold group, type the new group name on the top line and then press ENTER.

```

Setting cache data collection frequency

The Control DASD Display pop-up includes a field to control the frequency of cache data collection.
To display this pop-up, enter `l` (lowercase L) on the input field before the DASD indicator on the System Status panel.

```
<table>
<thead>
<tr>
<th>Workload status</th>
<th>Resource status</th>
<th>Operator Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ Batch --------</td>
<td>_ CPU --------</td>
<td>_ Key Task $$$$$</td>
</tr>
<tr>
<td>_ STC/APPC ------</td>
<td>1 DASD $$$$$</td>
<td>_ WTO Buffer -----</td>
</tr>
</tbody>
</table>

To change the frequency of data collection for this status light, specify a new rate (1 = every auto refresh, 2 = every other, etc.).

This light represents a composite status of all DASD groups, each supports thresholds for the resources below. To define thresholds for these resources use the Options menu Thresholds selection.

<table>
<thead>
<tr>
<th>Volume Collection Measures</th>
<th>Cache Collection Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device not responding</td>
<td>Cache read hit percent</td>
</tr>
<tr>
<td>Device dropped ready</td>
<td>Cache write hit percent</td>
</tr>
<tr>
<td>Device not using dynamic reconnect</td>
<td>DASD Fast write hit percent</td>
</tr>
<tr>
<td>Indexed VTOC lost</td>
<td>Volume cache inactive</td>
</tr>
<tr>
<td>Device response time</td>
<td></td>
</tr>
<tr>
<td>Device percent busy</td>
<td></td>
</tr>
</tbody>
</table>

F1=Help  F2=Keys  F3=Exit  F5=Refresh  F6=Console  F10=Action Bar  F11=Print
```

### Analyzing DASD problems

The Analyze DASD Problems panel reports volumes that have exceeded the thresholds set for cache.

To display this panel, enter `a` (Analyze Problems) on the input field of the DASD status indicator on the System Status panel.

### Finding inactive cache volumes

The DASD Response and % Busy panel reports the number of volumes that are eligible for cache but do not have cache activated.
To display this panel, enter **s** (Show Details) on the input field of the DASD status indicator on the System Status panel.

```
F1=Help    F2=Keys    F3=Exit    F5=Refresh    F6=Console   **=Bkwd   F8=Fwd
F10=Action Bar  F11=Print   F15=System Status
```

**Finding read and write hit percentages**

The DASD Details for a Device panel displays current cache read and write hit percentages and is accessed as follows.

- Enter **s** (Show Details) on the input field of the DASD indicator on the System Status panel.
From the DASD Response and % Busy panel, select a device by typing s in the input field.

To access the panel showing 3880 statistics, enter fast path iihc from any OMEGAMON II panel.

To access the panel showing 3990/2105 statistics, enter fast path iihs from any OMEGAMON II panel.

Viewing realtime cache statistics

Realtime cache statistics are reported separately for 3880-13/23 and 3990/2105 DASD cache controllers.

To access the panel showing 3880 statistics, enter fast path iihc from any OMEGAMON II panel.

To access the panel showing 3990/2105 statistics, enter fast path iihs from any OMEGAMON II panel.
This is a sample panel for a 3990-3 device.

### Viewing historical data for cache controllers

Historical details are available for the 3880-13/23 and 3990/2105 DASD cache controllers.

- To display historical details for 3880-13 and 3880-23 cache controllers, enter fast path `iihh` from any OMEGAMON II panel.
- To display historical details for 3990/2105 cache controllers, enter fast path `iihi` from any OMEGAMON II panel.

### Viewing historical cache statistics

Historical trends for cache statistics are reported similarly to the historical details for controllers described above.

- To access historical trends for cache statistics, enter fast path `iiht` from any OMEGAMON II panel.

### Example: Replacing Link List Datasets

**Introduction**

System administrators or other authorized users may want to update/replace link list datasets dynamically without performing an IPL. This example explains the process.

**Note:** This example applies only to users of OS/390™ Release 2 and below.

**Use with CAUTION!**
The utility invoked by the Replace Link List panel dynamically changes the system link list. The utility does not consider the tasks currently running on the system. To prevent library inconsistencies and possible failure of tasks that are running, perform the replacement with caution.

**Authorizations**

To perform a link list replacement you must be authorized to issue an OMEGAMON QLLA command. By default the security for this command is level three.

If you select the option to update the LLA directory, OMEGAMON II must be:
- authorized to issue the MVS START and STOP commands.
- given READ authority to SYS1.PARMLIB.

**Before you begin**

You will have to modify members IEASYSxx, LNKLSXxx, and/or CSVLLAxx in SYS1.PARMLIB before you begin a link list dataset replacement.

Refer to *MVS/ESA Initialization and Tuning Guide* and the *MVS/ESA Initialization and Tuning Reference* for more information.

**Viewing all link list datasets**

The Link List Datasets panel presents a list of datasets with their orders and APF-authorization statuses, as shown below.
To access the Link List Datasets panel, enter fast path `iksd` from any OMEGAMON II panel.

Replacing a link list dataset

Replacing a link list dataset is a two-step procedure. First you enter the suffix of the IEASYSxx member of SYS1.PARMLIB that will be used as input to the link list replace utility. To enter the suffix, enter fast path `iksr` to reach the Replace Link List Dataset pop-up window which prompts you for the two-character suffix.
Ensure that the IEASYS\text{xx} member exists before proceeding.

After entering the suffix on the Replace Link List Datasets pop-up, the Replace Link List panel appears. This is where you perform the second step in the procedure, entering the request to replace the current link list.

- To enter your request to replace the current link list with the proposed changes displayed on the panel, enter \textit{r} (for Replace) in the Library Link List field.
If the Library Lookaside Area (LLA) facility is installed at your site, a pop-up will appear that allows you to request the MVS STOP and START commands required to update the LLA directory with the new link list datasets.

Updates to the link list will take place after you confirm your request on a confirmation pop-up window.

Example: Resolving a Channel I/O Problem

Introduction
The following pages demonstrate how to use the OMEGAMON II Channel Activity panel to investigate a problem in the systems job flow.
Discovering the problem
You are monitoring your system from the System Status panel, and the Channels status light turns red.

To request more information for this indicator, issue the show details action for channels. To do this, position the cursor next to Channels, type S, and press Enter.
This will display the Channel Activity panel as shown in the following figure.
Investigating the problem

On this panel, you see that CHPID “84” is showing the highest activity. To obtain more information, enter / in the CHPID “84” input field.

This will display the Actions pull-down menu as shown in the following figure.

To determine the type of device attached to the CHPID, select option 1 from the pull-down menu.
If both TAPE and DASD devices are attached to the same CHPID, a pop-up will appear allowing selection of either device type. The Channel Activity panel with a pop-up will appear as shown in the following figure.

In this example, option 2 is selected. This will cause the Tape Drives panel to appear as shown in the following example.

To display device information, position the cursor on the input area in front of the desired device, type **s**, and press enter.
This will cause the Tape Unit Details panel to appear as shown in the following figure.

**Solving the problem**

In this example, the tape drive dropped ready. You could call Operations and have them ready the tape drive.
Investigating the problem

If you selected Option 1 (Display DASD attached to the CHPID) on the Channel Activity pop-up panel, then the DASD Response and % Busy panel would appear as shown in the following figure.

You notice the high response time for device number 9C3 of 255.3 ms. Enter S in the input field in front of device number 9C3. This will display the DASD Details for a Device panel as shown in the following figure.
Solving the problem

The high I/Os queue response time shows that there is a scheduling conflict for this device. This could be resolved by implementing I/O priority queueing or moving the datasets causing the scheduling conflict.

Example: Resolving a CPU Utilization Problem

Introduction

The following pages demonstrate how to use the OMEGAMON II to investigate a problem with CPU utilization.

Discovering the problem
You are monitoring your system from the System Status panel and you notice that the CPU indicator is red.

```
<table>
<thead>
<tr>
<th>Workload status</th>
<th>Resource status</th>
<th>Operator Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ Batch -----</td>
<td>_ STC/APPC -----</td>
<td>_ TSO: RTA -----</td>
</tr>
<tr>
<td>_ STC/APPC -----</td>
<td>_ DASD -----</td>
<td>_ Tape -----</td>
</tr>
<tr>
<td>_ TSO Host -----</td>
<td>_ Paging -----</td>
<td>_ Storage -----</td>
</tr>
<tr>
<td>_ Per1 TSO -----</td>
<td>_ Storage -----</td>
<td>_ Workload -----</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_ Workload -----</td>
<td>_ CSA -----</td>
<td>_ Channels -----</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F1=Help  F2=Keys  F3=Exit  F5=Refresh  F6=Console  F10=Action Bar  F11=Print
```

To request more information for this indicator, you issue the **Show Details** action for CPU. To do this, position your cursor next to CPU, type **S** and press Enter. This action displays the CPU utilization panel shown in the following figure.
Investigating the problem

The high CPU % and high TCB % shown above would indicate that test job TDM2S4E is in a LOOP condition.

Solving the problem
Cancel the job and the system performance will return to normal.

Example: Investigating a Key Task Alert

Introduction
The Operator Alerts section of the System Status panel provides information especially for operators of MVS systems as well as providing useful information for all OMEGAMON II users.

For example, the Key Task status indicator on the System Status panel gives the status of all critical jobs on your system. The light for Key Task becomes red or yellow if a critical job or a started task has stopped running unexpectedly or if a critical task was not started.
Example of a key task status alert

Suppose, for example, that Key Task has a red status light indicating a critical job or started task has a problem. (In the panel shown below, $$\text{dollar signs}$$ represents a red status light.) To find the reason for the alert, request the **Show Details** action for this object by typing **s** in the Key Task input field and pressing Enter.

![Key Task Alert window](image)

**Key Task Alert window**
The Key Task Alert pop-up window is displayed as shown below. It shows which critical tasks are missing.

![Key Task Alert Pop-up Window]

To restart a task or to see why it stopped, you must go to the system console. However, to go to the system console, you must have authorization.

**Note:** To obtain authorization to go to the system console, see your OMEGAMON II Customizer.

Upon obtaining authorization to go to the system console, press Enter from the Key Tasks Alert pop-up window. The system console is displayed.

### Example: Responding to an Enqueue Conflict

#### Introduction

As another example of an operator alert, we present an enqueue conflict. In this example, the Operator’s Toolkit, a special feature of OMEGAMON II, is introduced as a tool for handling an enqueue problem or other operator alert problems.

#### Example of an enqueue conflict

Suppose that a user has detected a “deadly embrace” because another job is vying for use of the same dataset and the user is not authorized to cancel the job. The user then calls the system operator.

The operator selects **Toolkit** from the Index pull-down, then **Operator’s Toolkit**. (The operator could also have used the fast path **iko**.)
To determine where the conflict is, the operator selects 3, Enqueue Conflicts and Reserves from the Operator’s Toolkit window as shown below.

![Enqueue Conflicts and Reserves panel](image)

The Enqueue Conflicts and Reserves panel appears, listing the enqueue conflicts currently existing. On this panel, the operator types / (slash) before the job to be cancelled.

Note: Warning: Be extremely careful when cancelling a job.
Once a job is selected, the Actions pull-down menu is displayed as shown below. The operator selects 6 to cancel the job.

<table>
<thead>
<tr>
<th>Actions</th>
<th>GoTo</th>
<th>Index</th>
<th>Options</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1. Show details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Bottlenecks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Historical trends</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Historical details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. Resource Details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6. Cancel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. Swap in</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8. Reset service class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9. Kill</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10. Exit F3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1=Help</td>
<td>F12=Cancel</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Toolkit for system programmers also**

**Note:** There is also a Toolkit for System Programmers in OMEGAMON II. It contains special system programmer functions that are not available through status alert lights. Like the Operator’s Toolkit, the System Programmer’s Toolkit is accessed from the Index pull-down; the fast path is **iks**.
Monitoring Performance across Systems

Section Overview

Introduction

Previous sections of this chapter described how to monitor the performance of a single system using OMEGAMON II panels. This section describes how to monitor more than one system at a time by:

- displaying realtime information from two or more systems on the same display screen using classic OMEGAMON commands
- displaying historical information about a shared-DASD device in a multi-system environment using classic EPILOG commands

**OMEGAMON and EPILOG and goal mode**

If you attempt to display service class or report class data through OMEGAMON while MVS is operating in goal mode, OMEGAMON may issue a message informing you that certain fields or an entire display is not applicable in goal mode. If you request a report through EPILOG that spans a time period during which WLM-based data was being collected, no data will be available for the report.

In this section

In this section we cover the following topics.

- “Monitoring Realtime Performance across Systems” on page 307
- “Monitoring Historical Shared-DASD Performance” on page 311

Monitoring Realtime Performance across Systems

**Introduction**

You can display realtime information from two or more systems on the same display screen through the Cross Memory (XMF) and Cross System (XSF) facilities. These facilities enable you to enter commands for, and receive information from, the following Candle products on one OMEGAMON display screen:

- OMEGAMON for MVS
- OMEGAMON for CICS
- OMEGAMON II for DB2 (XMF only)
- OMEGAMON for IMS
- OMEGAMON for VM (XSF only)

**How is this accomplished?**
One of the products listed above, typically OMEGAMON for MVS, acts as a director running in either dedicated or VTAM mode. The director controls the display screen and communicates with the other OMEGAMON products by passing your commands to the appropriate product acting as a collector. The collector passes the output of the commands back to the director, which in turn displays the output on its display screen.

For the sake of simplicity, the examples presented here will focus on using an OMEGAMON for MVS director in dedicated mode.

**Cross memory/cross system modes**

A single director can communicate with up to seven OMEGAMON collectors. You have a choice of mode for each collector:

- **cross memory mode**
  
  Collects information from OMEGAMON products that are running on the same MVS system. This is useful if you want to limit the number of dedicated terminals you have or provide a single point where you can monitor MVS, IMS, and CICS regions.
  
  In this mode, the director communicates with the collectors through a window in the Common Service Area (CSA).

- **cross system mode**
  
  Collects information from OMEGAMON products that are running on entirely different MVS operating systems, even if those systems are running at different levels of MVS.
  
  In this mode, the director communicates with the collectors through a dataset that is accessible by both.

A director can communicate with collectors in both modes at the same time and on the same director display screen, enabling you to pull together OMEGAMON displays from all over your site.

**Preparing to use XMF and XSF**

Before you begin using the cross memory and cross system facilities, you must start a director and the desired OMEGAMON collectors. Check with your OMEGAMON II Customizer to see if:

- the cross system and cross memory facilities have been installed according to the instructions provided in the *OMEGAMON II for MVS Configuration and Customization Guide*.
- a dedicated 327x terminal is available
- the appropriate director and collectors have been started

Once started, collectors running in **cross memory mode** automatically appear on your display terminal. If you plan to use a collector running in **cross system mode** (one that is gathering information from an OMEGAMON on another MVS system) you must first use the `/ATTACH INFO-line command to identify to the director which dataset is to be used for cross system communication; information gathered by this collector will not appear on your display terminal until this step is performed.
Displaying XMF/XSF information on a screen

When the XMF/XSF director and collectors have been started, the screen display on the dedicated 327x terminal divides into segments to reflect activity for the director and each collector simultaneously. The director segment appears first, followed by segments for up to seven collectors.

An INFO-line shows where each segment begins. You can type OMEGAMON commands on each of the INFO-lines; when you press Enter, the commands are executed by the respective collectors and their results displayed in the appropriate segments.

Example of an XMF/XSF screen

The following is an example of an XMF/XSF screen reflecting an OMEGAMON for MVS and OMEGAMON for CICS session running on CPU A033, and an OMEGAMON for IMS session running on CPU IPRD.

```
__________ #01       DIR      OM/DEX   V750./C A033   mm/dd/yy 16:58:45
... ...
OMEGAMON/MVS commands (running on A033)
...
__________ #01       XMM      OM/CICS  V520.00 CPRD   mm/dd/yy 16:58:45
... ...
OMEGAMON for CICS commands (running in the CICS production region)
...
__________ #01       DSK      OM/IMS   V500.99 IPRD   mm/dd/yy 16:58:45
... ...
OMEGAMON/IMS commands (running on IPRD)
...
```

In this example, the screen contains three INFO-lines that identify the start of a screen space dedicated to each of the three monitors. Each segment is identified by the 4-character system ID, which immediately precedes the date. In this case, the first OMEGAMON segment is identified by A033 (which represents the SMF ID for this system), the OMEGAMON for CICS segment is identified by CPRD, and the OMEGAMON for IMS segment by the ID IPRD.

Each segment also has a 3-character field to the right of the screen space identifier that denotes which type of segment it is:

- **DIR** The director segment.
- **XMM** A cross memory mode segment.
- **DSK** A cross system mode segment.

Controlling the XMF/XSF screen

Each cross memory/cross system segment acts like a normal OMEGAMON screen. You can perform any of the following actions within a segment, independently from the other segments:

- enter OMEGAMON commands
- scroll up and down
use function keys

ring the terminal bell (note, however, that in this case there is no indication as to which segment was responsible)

You can further control the XMF/XSF screen as follows:

- When displaying information from an OMEGamon that supports extended color and highlighting capabilities, the collector recognizes only the first color encountered for each line.
- If you press the CLEAR or PA1 key, an attention is propagated to each collector segment in a process that may take one or two cycles to complete. (This is useful if you accidentally turn on extended color for a terminal that is not equipped for extended color support, and a reset is required.)
- You can control the number of lines displayed for each collector segment by using the /GIVE and /TAKE INFO-line commands to transfer lines from one segment to another. See the OMEGamon for MVS Command Language Reference Manual for information on these commands.

**Synchronizing cross memory collectors**

In cross system mode, the collectors run on a different system from the director, so they communicate with each other through a dataset which resides on a DASD volume shared by both systems. This dataset acts as a temporary repository for commands and data, thus freeing the director from waiting for the collectors for an answer.

In cross memory mode, on the other hand, a director communicates with a collector through a window in the MVS Common Service Area (CSA), must wait for a response from the cross memory collector. The director attempts to synchronize with each cross memory collector so that each collector responds immediately to new commands that you issue; any interval-setting commands such as .SET INTERVAL are ignored. The director waits for each collector only for as long as the director’s cycle time (five seconds by default). If a collector fails to respond in this interval, the director stops waiting for that collector and goes on to the others.

When a collector falls out of sync, any commands you enter within its segment take one cycle to process, instead of being handled immediately. You will also notice a time difference between the time displayed on the collector’s INFO-line and that of the director.

If a collector can respond within the director’s cycle time but is very slow, it causes the director to slow down as well. You might find this condition with a low-priority CICS or IMS test region. You can eliminate this problem by using the /WAIT OFF INFO-line command in the problem segments to tell the director not to wait for these slow collectors. To synchronize the director and the collector again at any time, use the /WAIT ON INFO-line command. See the OMEGamon for MVS Command Language Reference Manual for information on the /WAIT command.

**Do directors act differently in VTAM mode?**

This section assumes your OMEGamon II Customizer has installed the cross memory and cross system facilities in dedicated mode. If XMF and XSF have been installed in VTAM mode, you will see the following differences:
Monitoring System Performance

- If you press a PF key while the screen is updating, your keyboard will lock. To correct this situation, press the RESET key.
- INFO-line commands do not process until you press Enter.
- The INTERVAL= parameter of the .SET command displays and changes the update interval.
- The .AUPOFF and /AUPOFF commands are ignored by the VTAM mode director.

Monitoring Historical Shared-DASD Performance

Introduction
While monitoring your system, suppose you notice that workloads are often waiting for I/O to a particular device. You know that this device is shared by more than one system, and you would like to see how it has been used by workloads and performance groups on the other systems to help you decide how to alleviate the I/O problems on your system.

The following paragraphs describe how to use the JDAS keyword of the EPILOG DISPLAY command to display historical information about a shared-DASD device in a multi-system environment.

Procedure
To show the cross-system impact of workload utilization against a particular shared-DASD device, use the following procedure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Make sure your OMEGAMON II Customizer has defined all systems to be monitored to the EPILOG reporter session.</td>
</tr>
<tr>
<td>2</td>
<td>Type ge in the action bar input field of an OMEGAMON II panel and press Enter to prepare to enter EPILOG commands.</td>
</tr>
<tr>
<td>3</td>
<td>Enter the DISPLAY command with the JDAS keyword. Supply the volser and appropriate time period. See the EPILOG for MVS Command Language Reference Manual for information about the EPILOG JDAS keyword.</td>
</tr>
<tr>
<td>4</td>
<td>When you have completed your shared-DASD investigation, type end on the EPILOG command line and press Enter to return to OMEGAMON II.</td>
</tr>
</tbody>
</table>

Example: shared-DASD degradation by workload
Suppose you have determined that there may be cross-system contention on volume COM002. You enter the following EPILOG DISPLAY command to display degradation by workload between 2:15 and 2:30 PM yesterday:

```
DISPLAY JDAS(COM002) YDAY STIME(14:15) ETIME(14:30)
```
The following report displays:

<table>
<thead>
<tr>
<th>Workload_JES #_Sysid_Time</th>
<th>%</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASMPBKUP J4661 SYSA 5:32 M 82.1</td>
<td>---------------</td>
<td>&gt;------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AOASMP2 S4533 SYSF 45:98 S 11.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDNY30 T4014 SYSA 26:22 S 6.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workload_JES #_Sysid_Time</th>
<th>%</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$AOASMP1 S4243 SYSF 29:64 S 66.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASMPBKUP J8812 SYSA 13:94 S 31.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AOASMP2 S4533 SYSF :84 S 1.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this figure, you can see both the delayed and the heavily active workloads accessing the volume on a comparative basis. For instance, job $AOASMP1 on SYSF was queued for I/O 66.7% of the time, while job ASMPBKUP on SYSA spent 82.1% of its time during the interval actively performing I/O.

The Active I/O and Queued I/O status percentages each add up to 100%. These percentages reflect the portion of the total observed active or queued I/O against volume COM002 that is attributable to each job.

In other words, the number of samples for active I/O against COM002 during the time period 2:15 to 2:30 is accumulated from all eligible records on each datastore in the active datastore list. Each sample is weighted according to the sampling interval time in effect for each system involved.

In addition to the total accumulation, an accumulation for each job is also performed. The percentages are then calculated by dividing the job within system accumulations by the total accumulation across all jobs in all systems. This calculation is applied separately to active, queued, and reserved I/O wait reasons.

Recommendations

You can use the information provided in the JDAS display to help you decide how to alleviate I/O problems on a system. If you have issued the DISPLAY command with different time periods and you see a trend in DASD use among systems, you might take one of the following actions to alleviate device contention:

- allocate datasets needed by jobs on different systems to different devices
- spread workloads among systems more evenly
Introduction

When the applications that run on your system are not executing at peak efficiency, the resulting delay in service can set off a chain reaction of missed deadlines and reduced productivity.

This chapter describes how to analyze workload performance on your system by:

- explaining bottleneck and impact analysis
- describing their associated controls and settings
- presenting a scenario that shows how to resolve a typical bottleneck
- providing general guidelines for handling common bottleneck situations

Chapter Contents

Understanding Bottleneck and Impact Analysis ........................................... 314
Collecting Bottleneck and Impact Analysis Information .......................... 316
Excluding Wait Reasons from Bottleneck Analysis ................................. 318
Example: Diagnosing a Current Bottleneck .............................................. 320
Example: Analyzing a Recurring Bottleneck ........................................... 328
Guidelines for Troubleshooting Bottlenecks .......................................... 333
Understanding Bottleneck and Impact Analysis

Introduction

OMEGAMON II surveys system resources to determine where your workloads are spending their time, and helps you see if they are spending that time productively or unproductively; it does this by reporting whether they are active or waiting for a resource. If a workload is waiting for a resource, OMEGAMON II also identifies the impact that other workloads may be having on the workload.

What is bottleneck analysis?

A bottleneck is a condition that causes a workload (a batch job, started task, or TSO session) to spend its time unproductively, preventing it from reaching its performance goal. In helping you identify bottlenecks, OMEGAMON II breaks down the elapsed or response time of a workload into the following execution states. These execution states identify the current activity of a workload.

<table>
<thead>
<tr>
<th>IF the execution state is...</th>
<th>THEN the workload is...</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>actively using the CPU or performing I/O.</td>
</tr>
<tr>
<td>idle</td>
<td>voluntarily inactive. This includes swap conditions that the workload issues for itself when it expects to be waiting for a long period of time.</td>
</tr>
<tr>
<td>delayed</td>
<td>prevented from continuing its processing because it is waiting for a resource.</td>
</tr>
</tbody>
</table>

OMEGAMON II considers workloads to be in a bottleneck situation when their productivity is degraded because they are waiting for resources.

What causes a workload delay?

Wait reasons explain why the execution of a workload is delayed. Examples of wait reasons include:

- waiting for CPU
- swapped out to improve central storage availability
- waiting for I/O to a device

Can an active workload be constrained by a bottleneck?

A workload does not have to be waiting for a resource to be constrained by a bottleneck. Although a workload that is actively using the CPU or performing I/O is usually considered to be executing productively, its performance may be degraded if the amount of time it spends performing these activities is excessive due to the activity of another workload. Thus, even an active workload can be in a bottleneck situation.
From bottleneck analysis to impact analysis

Once OMEGAMON II determines the execution state of a workload, it identifies other workloads that are impacting the workload by contending for the same resources. Once you know the source of the resource contention, you can analyze the contention between the workloads and take appropriate action.

Displaying bottleneck information in OMEGAMON II

The OMEGAMON II Bottlenecks panel contains bottleneck and impact analysis information. There are four ways to access this panel:

- Enter the \texttt{b} action code next to a problem workload.
- Select \texttt{Bottlenecks} from the Actions or GoTo pull-downs where available.
- Select \texttt{Resource Bottlenecks} from the Actions pull-down for a wait reason on the Details for a Job or Started Task panel. These detail panels list the execution states of the address space; you can press F1 on an execution state for recommendations on where to turn next.
- Enter fast path \texttt{iww} to display the Workload Manager Realtime menu. Then enter one of the following mnemonics to display a panel from which you can request bottleneck analysis.
  - \texttt{c} for Report Class Address Spaces
  - \texttt{g} for Resource Group Details
  - \texttt{a} for Service Class Address Spaces
  - \texttt{i} for Service Class Period Address Spaces
  - \texttt{k} for Service Class Period Workflow Analysis
  - \texttt{p} for Service Class Periods
  - \texttt{f} for Service Class Workflow Analysis
  - \texttt{v} for Service Classes Served by an Address Space
  - \texttt{o} for Workload Manager Overview

Resolving the bottleneck

Once you have identified the cause of the bottleneck, you can resolve the situation in a number of ways. For guidelines on resolving bottlenecks, use the online help system on the Bottlenecks panel and read “Guidelines for Troubleshooting Bottlenecks” on page 333.

Multi-tasking workloads

Some workloads are capable of performing in multiple execution states simultaneously. An example is a workload that is actively using CPU while delayed waiting for a tape mount. Such workloads are considered multi-tasking and, therefore, capable of achieving resource total percentages greater than 100%.
Collecting Bottleneck and Impact Analysis Information

Introduction
By default, OMEGAMON II collects enough bottleneck and impact analysis information to solve short term, immediate problems. This section explains how to override this default collection interval if necessary.

Solving current problems
OMEGAMON II begins collecting bottleneck and impact analysis information automatically when you display the Bottlenecks panel, and (by default) stops collecting information when you leave the Bottlenecks panel. This default collection interval provides the information you need to resolve a current bottleneck problem.

Solving recurring problems
If bottleneck situations occur frequently during the day, or if the execution of a particular workload is frequently delayed, you may want to watch these situations over a longer period of time. The Session Defaults pop-up enables you to broaden the collection interval, activating bottleneck and impact analysis even while you’re using other panels, up to the duration of your OMEGAMON II session.

Implications of continuing collection
There is a significant amount of overhead associated with collecting bottleneck and impact analysis information. Therefore, we recommend that you keep the defaults of Off and No (for bottleneck and impact analysis, respectively) for resolving most short-term bottleneck problems, and change the defaults only when long-term bottleneck situations occur.

Session Defaults: fast path
The fast path for setting session defaults is ocs.

Session Defaults pop-up
The following figure is an example of the Session Defaults pop-up. We will limit our discussion of the session defaults you can specify to those related to bottleneck and impact analysis:

- bottleneck analysis
Collecting Bottleneck and Impact Analysis Information

**Bottleneck and impact analysis defaults**

The default entries on the Session Defaults pop-up for bottleneck and impact analysis are Off and No, respectively. As previously explained, this means that collection of bottleneck information will not begin until you display the Bottlenecks panel, and will end when you leave the Bottlenecks panel.

To troubleshoot long-term bottleneck problems, you must change these settings to On and Yes. This causes collection of bottleneck and impact analysis information to continue after you have exited the Bottlenecks panel, until you turn collection off or end your OMEGAMON II session.

You can turn collection off at any time during your OMEGAMON II session by returning to the Session Defaults pop-up and restoring the default entries for bottleneck (Off) and impact (No) analysis.

---

**enable impact analysis**
Excluding Wait Reasons from Bottleneck Analysis

Introduction

Some wait reasons may be of more interest to you than others when you are performing bottleneck analysis on a workload. OMEGAMON II enables you to specify the wait reasons that you would like to exclude from bottleneck analysis through a selection on the Options pull-down.

Excluding wait reasons: fast path

You select the wait reasons you want to exclude using the Wait Reason Reporting Control pop-up; the fast path is ocw.

Types of wait reasons to exclude

The Wait Reason Reporting Control pop-up enables you to exclude any of the following types of wait reasons:

- CPU waits
- enqueues
- idle waits
- paging waits
- SRM delays
- tape mounts
- other waits

OMEGAMON II always monitors and reports workloads that are actively using CPU, actively performing I/O, or queued to perform I/O. Therefore, you will not see selections for these execution states on the Wait Reason Reporting Control pop-up.

Rules for selecting wait reason types

By default, all wait reason types are selected. To exclude a wait reason type from being reported, erase the slash (/) next to the wait reason type.

If you exclude one or more wait reason types, then for the period of their exclusion, OMEGAMON II will confine its wait reason reporting to the remaining wait reason types. For instance, if you exclude tape mount delays and the actual amount of time the workload spent waiting for a tape mount was 20%, OMEGAMON II recomputes the other wait reason values to absorb that 20%.

Example: Wait Reason Reporting Control pop-up

The Wait Reason Reporting Control pop-up is shown below. In this example, all wait reasons have been selected for reporting except for enqueues and idle waits.
For more information

For a list of the specific wait reasons that are associated with each wait reason type, and for more information that may help you decide whether to exclude a wait reason type or not, press F1 next to the wait reason type.
Example: Diagnosing a Current Bottleneck

Introduction
This scenario shows how to use OMEGAMON II to resolve a current resource contention problem that is affecting a CICS workload.

Discovering the problem
Suppose you are monitoring your system from the System Status panel and the Workload light turns yellow. It stays yellow for a few minutes and then turns red. You immediately request more details by positioning your cursor to the left of the Workload indicator and pressing Enter.

OMEGAMON II displays the Workload Manager Overview panel, as pictured on the following page.

Finding the problem service class
The Workload Manager Overview panel displays WLM performance statistics for period 1 of all active service classes. The last column of the display shows the status of the worst period of periods 2 - 8 for each service class.

Service class CICSSHIP has a performance index (P/I) of 1.5 and an importance level of Highest. At this point, you want to see more information on the address spaces that serve CICSSHIP.
Example: Diagnosing a Current Bottleneck

You know that you entered all address spaces that serve CICSSHIP in a special service class, SERVSHIP. To view information on these address spaces, you enter fast path \texttt{iwwf}, and enter SERVSHIP when prompted for the service class name.

If you want to see statistics on workload resource consumption, press the \texttt{<WLM Resource Utilization>} pushbutton.

**Examining resource contention**

Entering fast path \texttt{iwwf} results in the Service Class Workflow Analysis panel pictured below. This panel provides information on resource contention for all address spaces in service class SERVSHIP.

![Service Class Workflow Analysis Panel]

It is clear from the display that I/O contention is the main bottleneck for address spaces in service class SERVSHIP.

To find the volume on which the I/O contention is occurring, you request details by moving the cursor to the I/O wait input field and pressing Enter.

**Examining resource utilization**

Entering fast path \texttt{iwwu} results in the Workload Manager Resource Utilization panel pictured below. This panel provides information on resource utilization for all service classes under the current Workload Manager service policy.
If the status of a service class reflects a warning or critical condition, you can select the service class to display details about its periods.

**Locating the volume**

Requesting more details for I/O wait on the Service Class Workflow Analysis panel leads you to the Bottlenecks panel pictured below.
Example: Diagnosing a Current Bottleneck

This display shows you that the majority of the I/O contention is occurring on volume LRG010.

To continue the diagnosis of this problem, move the cursor to the input field for the queued state of LRG010 and press Enter.

Identifying the type of contention

Requesting more details for a volume on the Bottlenecks panel results in the DASD Details for a Device panel pictured below.

This panel shows you a high IOS queue and Disconnect time for volume LRG010.
You know that the high disconnect time can be caused by seeking on the volume. To obtain seek information for volume LRG010, you move the cursor to the Seek Analysis pushbutton and press Enter.

**Identifying the greatest impactor**

The pushbutton on the DASD Details for a Device panel leads you to the DASD Seek Analysis for a Device panel pictured below.
You recognize SHIPAOR1 as a server address space in the SERVSHIP service class. It is in contention with job D60BRP1B for volume LRG010.

To obtain more details on contending job D60BRP1B, you place the cursor on the input field to the left of @@@@ = D60BRP1B and press Enter.

More about the greatest impactor

The Details for a Job or Started Task panel appears, as pictured below.
Example: Diagnosing a Current Bottleneck

This panel shows you that D60BRP1B is a batch job submitted by a TSO user. Since a batch job is much less important than a CICS server, you decide to quiesce job D60BRP1B.

To reach the panel where you can quiesce the job, you enter fast path **iwwa** in the Action bar input field, and enter TSOBATCH when prompted for the service class name. The Service Class Address Spaces panel (page 327) appears.

### What does quiesce do?

When you issue a *quiesce* request, SRM swaps the job out if it is swappable, or reduces its dispatch priority to least favorable if it is non-swappable. You can issue a resume request at a later time to allow the batch job to run when the server is less dependent on this volume, or you can cancel the job.

### Quiescing the impactor

The Service Class Address Spaces panel shows all the address spaces in the TSOBATCH service class. You see job D60BRP1B in the first column.

You type a slash (/) to the left of D60BRP1B, and enter an `a` in the Action bar input field. OMEGAMON II displays the Actions pull-down menu for this panel, as shown below.
You enter 4 on the Actions pull-down menu, to quiesce job D60BRP1B.
Later, when you show details for CICSSHIP, its performance has improved dramatically.
Example: Analyzing a Recurring Bottleneck

Introduction

This scenario shows how to use OMEGAMON II to analyze a recurring workload delay that is being caused by a resource group maximum.

What is a resource group maximum?

A resource group maximum is a WLM setting that limits the total amount of CPU resources that a group of work can use, thereby protecting that group from consuming more resources than its budget allows. This can occur during sudden bursts of computer activity, when a job gets caught in a program loop, and so on. A resource group maximum is also referred to as a resource group cap.

Background information

You work for a company that sells computer resources. You assign all of a client’s work to a single resource group. You can then use the resource group maximum to limit the amount of computer resources available to each client. There are two reasons that you limit a client’s resources:

- The client might request the limit to prevent excess consumption.
- You might need to limit one client’s use of resources in order to guarantee enough resources to satisfy other client’s service-level agreements.

Discovering the problem

Your department receives a complaint from ABC Incorporated regarding the time it takes a particular batch job to complete. The customer maintains that in the past he submitted the job sometime in the early afternoon, and it completed by 3:00 or 3:30 that day.

Recently, though he submits the job at the same time, it continues to run until after the last person leaves the office at 5:30 or 6:00.

Selecting a date and time range

You want to examine the job’s performance during one of its recent runs. According to the customer, batch job ABCWKSUM was running Friday afternoon from 1:00 to sometime after 5:30.

To set the historical time period for the historical detail panels, you enter fast path ohh from any panel. The Hist. Detail Date Time Settings pop-up window appears as pictured below.
Example: Analyzing a Recurring Bottleneck

You enter start and end times for the job and Friday’s date as the start and end dates. You press Enter and then F12 to return to the original panel.

Examining the job’s performance

Now that the historical time period is set, you can proceed to the Historical Details for a Batch Job, STC, or TSO User panel. To reach this panel, you enter fast path iwgc and enter ABCWKSUM when prompted for the batch job name. The following panel appears.
This panel provides two clues: during the 5-hour period Friday afternoon, batch job ABCWKSUM was delayed 40% of the time because the resource group had reached its maximum and 50% of the time waiting to use the CPU.

**Interpreting delay statistics**

When a resource group reaches its prescribed maximum, MVS stops all work in the group to allow the group’s CPU usage to fall below the maximum. OMEGAMON II refers to this as a “resource group capping” delay. You therefore know that this job’s resource group reached its maximum during much of the 5-hour period, since ABCWKSUM was delayed for this reason 40% of the time.

ABCWKSUM was also unable to execute regularly when the group was not in the capping delay state as evidenced by a CPU wait of 50%. You suspect that a low WLM performance goal, or possibly none at all, is assigned to the job’s service class.

To display the goal of service class ABCBATCH for the active time period, you enter fast path `iwmh` and enter ABCBATCH when prompted for the service class name.

**Examining the service class goal**

OMEGAMON II displays the Historical Details for a Service Class panel pictured below.

This display explains the poor service that the job is receiving. A performance goal of Discretionary increases the possibility that work in other service classes in the resource group will receive computer resources before jobs in service class ABCBATCH. The 50% CPU wait is probably the result of this performance goal. You would like to see usage statistics for other service classes in the resource group that were active during that time period, so you move the cursor to the Resource group input field and enter h.
Examining the resource group

OMEGAMON II displays the Historical Details for a Resource Group panel pictured below.

This panel confirms that, collectively, the active service classes in the resource group reached the maximum CPU limit during much of the 5-hour period. The % Max. Capacity Used statistic on this panel is an average for the time period, so a value of 97% means that the resource group was hitting its limit during much of Friday afternoon. This substantiates the capping delay you saw on the Historical Details for Batch Job, STC, or TSO User panel.

You recognize the other service classes in this resource group as online production applications, which would have service-level agreements (performance goals and importance levels) that guarantee them more CPU resources than is guaranteed to ABCBATCH.

Why poor service now?

It is now clear to you why job ABCWKSUM is receiving poor service. However, there is still one outstanding question in your mind, and you know it will be in your client’s as well. Why is this job taking so much longer to process now than it did in the recent past? Luckily, you see a possible answer on the current display.

You do not recognize service class ABCPERS, and it is consuming a significant share of the resource group’s capacity. So, you examine the client service log for ABC Incorporated and discover that a new personnel application went into production 2 weeks ago for ABC Inc. The log explains the last outstanding question.
Problem resolution

You call the client and inform him of the new personnel application, which was using about 32% of the maximum CPU capacity last Friday afternoon. He now understands the recent change in performance for ABCWKSUM.

You suggest he review the contract with your contract supervisor if he decides that his company requires more computer resources than those specified in the original contract.
Guidelines for Troubleshooting Bottlenecks

Introduction

The execution states reported by OMEGAMON II fall into two categories: waiting for resources and actively using resources.

The tables in this section describe general symptoms of these two categories of bottlenecks and their possible causes, and suggest actions you might take to alleviate bottleneck situations.

Discovering symptoms

You can discover the bottleneck symptoms described in the following tables by:

- observing the status indicators on the System Status panel and navigating to more detailed panels from there (such as the Details for a Job or Started Task panel or the Analyze Problems or Bottlenecks panels)
- receiving a call from a user reporting an address space (batch job, started task, or TSO user) that isn’t performing as expected and displaying details for that address space

Waiting for resources

The following table describes common situations in which waiting for a resource might become a bottleneck that degrades the performance of a workload. You can recognize these situations by looking at information gathered collectively from the Bottlenecks, Resource Bottlenecks, and Details for a Job or Started Task panels.

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Possible Causes</th>
<th>Possible Solutions</th>
</tr>
</thead>
</table>
| waiting for I/O    | wait for device          | move datasets to another DASD volume
|                    |                          | reschedule either job
|                    |                          | identify the competing workloads and swap them out
| enqueue delays     | enqueue for datasets     | release enqueue of less important job by canceling the job          |
| paging waits       | real storage contention  | adjust the service class goal                                        |
|                    | I/O subsystem is         | reschedule the job                                                  |
|                    | constrained              | add a page dataset                                                  |
|                    | device contention        | ensure there are no other active datasets on page devices           |
## Actively using resources

The following table describes common situations in which actively using a resource might become a bottleneck that degrades the performance of a workload. You can recognize these situations by looking at information gathered collectively from the Bottlenecks, Resource Bottlenecks, and Details for a Job or Started Task panels.

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Possible Causes</th>
<th>Possible Solutions</th>
</tr>
</thead>
</table>
| SRM delay                              | swaps performed to accommodate other address spaces | ■ make impacted workload nonswappable  
■ select another workload to swap out  
■ adjust the service class goal         |
| waiting for CPU                        | looping condition                    | reset the service class goal  
reduce the impact of the other address space by canceling it or QUIESCEing it |
| waiting for CPU                        | overcommitted CPU                    | reset the service class goal  
reduce the impact of the other address space by canceling it or QUIESCEing it |

### Symptom

- unusually high or low CPU use
  - looping condition
  - inefficient coding
  - select **Inspect CPU utilization** from the Actions or GoTo pull-downs to see if program is written inefficiently or contains a loop

- active I/O
  - inefficient placement of datasets on device
  - disk fragmentation
  - on non-SMS volumes, move concurrently used datasets closer together
  - on SMS-managed volumes, set dataset to must-cache status
  - reallocate or compress a dataset
  - move dataset to a cache device
Introduction

OMEGAMON II realtime panels display your system’s performance problems as they occur. OMEGAMON II also provides historical information about your system’s workloads and resources. In this chapter, we describe how to use OMEGAMON II to gain historical perspective and identify trends.

Chapter Contents

Overview ................................................................. 336
Setting Controls for Online Historical Panels ......................... 337
Displaying Historical Detail Information .............................. 338
Displaying Historical Trend Information .............................. 343
Overview

By studying the information gathered from an historical perspective, you can:

- determine whether current performance is standard or problematic
- identify trends

You can display WLM-based historical detail and historical trend information online using action codes and pull-down menus from any OMEGAMON II panel.

In this chapter, we describe how to use these action codes and pull-down menus, and how to set controls to select the most appropriate detail and trend information.

What are historical details?

Historical detail panels provide information for a specified time period. For instance, if you notice a current DASD problem, and you would like to see how long it has been occurring, you can display a historical DASD panel showing details for the previous three hours. The resulting panel combines historical DASD information into a single averaged result.

What are historical trends?

In contrast to historical detail panels, historical trend panels show trends over a series of time intervals. For instance, to expand on the DASD historical details example described above, you might display a historical DASD trend panel to look at DASD activity during the same time period, but for every day for the past two weeks.

MVS/SP 5 and historical data

When navigating to an historical panel, you sometimes receive a message indicating that no records were found for the time period you requested. Under MVS/SP 5, this can mean that MVS was running in compatibility mode during the requested period and only IPS-based data is available.
Setting Controls for Online Historical Panels

Introduction
The following controls help you select the most appropriate information for your historical detail and trend panels:
- history refresh interval
- date and time range for trend panels
- date and time range for detail panels

These controls are described below.

Setting history refresh interval
Historical data is typically collected to a historical datastore at the end of every RMF interval. This data is not available for historical reporting in OMEGAMON II until a refresh is performed. A refresh is the frequency with which OMEGAMON II checks the historical datastore for newly collected data.

The default refresh interval is 60 minutes. If the RMF interval on your system is set to 15 minutes, this means that a refresh will occur every fourth RMF interval, which ensures that no more than four RMF intervals will elapse before historical data is available to OMEGAMON II.

The value you choose as a refresh interval depends on how current you want the data in your historical reports to be. Although a refresh interval shorter than 60 minutes requires more overhead than one which is longer, it ensures more current data. It’s up to you.

To set the history refresh interval for historical panels:
- select Historical Online/Batch Controls from the Options pull-down
- then select History Refresh Interval Selection (or enter fast path ohr).

Setting detail date and time
To set a date and time range for historical detail panels:
- select Historical Online/Batch Controls from the Options pull-down
- then select History date/time selections (or enter fast path ohh).

Setting trend date and time
To set a date and time range for historical trend panels:
- select Historical Online/Batch Controls from the Options pull-down
- then select Trend date/time selections (or enter fast path oht).
Displaying Historical Detail Information

Introduction

You can display panels containing historical detail information by using action codes and pull-down menus.

Access methods

There are three ways to display historical detail panels.

<table>
<thead>
<tr>
<th>Access Method</th>
<th>Is Available From</th>
<th>And Results In</th>
<th>Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical details</td>
<td>any workload or resource light on the System Status panel,</td>
<td>historical details about a workload or resource for a specified time period.</td>
<td>h</td>
</tr>
<tr>
<td>action code</td>
<td>or on a detail line of subsequent panels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GoTo pull-down</td>
<td>action bar</td>
<td>selections for related historical panels</td>
<td>g</td>
</tr>
<tr>
<td>Index pull-down</td>
<td>action bar</td>
<td>a list of historical detail categories</td>
<td>ih</td>
</tr>
</tbody>
</table>

We will discuss each access method in turn.

Example of H action code

One way to display a historical detail panel is to use the H action code.

Suppose you are monitoring your system from the System Status panel and you see the Workload status indicator turn red. You request the Show details action for this indicator to display the Workload Manager Overview panel, as shown in the following figure.

When the Workload Manager Overview panel appears, you notice there are currently two service classes that have rather high performance indexes (P/I) for period 1.
You wonder how long service class TSOPROD1, with an importance rating of Highest and a performance index of 1.92, has been running at almost twice its average response time goal of 350 milliseconds.

To answer this question, you need historical details for period 1 of service class TSOPROD1. Using fast path **ohh**, you request a date and time range that spans the previous hour.

Then, you enter **h** next to service class TSOPROD1, as shown above, to request historical details.

Entering **h** on the Workload Manager Overview panel leads you to the Historical Details for a Service Class panel illustrated in the following figure.
Displaying Historical Detail Information

The level of historical detail on this panel shows you that period 1 of service class TSOPROD1 has been achieving its goal for at least an hour as indicated by a performance index of .94 for that period.

Since the goal for the service class period has on average been achieved over the previous hour, the red light most likely does not indicate an unrealistic MVS performance goal or OMEGAMON II threshold for this service class period. You will not make adjustments in either of these areas.

However, if you want to further investigate the current poor performance, you can display workload analysis information for the service class. To do so, move the cursor to the <Svc Cls Wkflw Analysis> pushbutton at the bottom of the current panel, and press Enter.
Example of GoTo pull-down

Another way to display a historical detail panel is to use the GoTo pull-down menu. In this example, displayed by entering <g>, eleven categories are available for navigation.

If you choose selection 8 from this menu, OMEGAMON II displays historical detail information for overall system performance.
Example of Index pull-down

A third way to display a historical detail panel is to use the Index pull-down menu. In this example, displayed by entering fast path ih, three categories of historical detail panels are available — workloads, I/O, and system.

When you select a category from this menu, OMEGAMON II displays another menu containing a list of the historical detail panels for that category.
Displaying Historical Trend Information

Introduction

You can display panels containing historical trend information by using action codes and pull-down menus.

Access methods

There are three ways to display historical trend panels.

<table>
<thead>
<tr>
<th>Access Method</th>
<th>Is Available From</th>
<th>And Results In</th>
<th>Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical trend</td>
<td>workload or resource lights on the System Status panel, or on a detail line of</td>
<td>historical trends about a workload or resource over a series of time intervals.</td>
<td>t</td>
</tr>
<tr>
<td>action code</td>
<td>subsequent panels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GoTo pull-down</td>
<td>action bar</td>
<td>selections for related historical panels</td>
<td>g</td>
</tr>
<tr>
<td>Index pull-down</td>
<td>action bar</td>
<td>a list of historical trend categories</td>
<td>it</td>
</tr>
</tbody>
</table>

We will provide an example of each access method.

Example of T action code

Suppose you receive a phone call from a group of TSO users complaining that response time is poor between 3:30 and 4:00 PM every day. You decide to take a look at the response times for the group’s service class, TSODEV, over the past week or so.

To access TSO service classes, you enter s next to the Per1 TSO status light on the System Status panel.

Entering s on the System Status panel displays the TSO version of the Workload Manager Overview panel pictured below.
You scan the first column, looking for the service class of the TSO development group that complained of the slow response time.

When you locate service class TSODEV, you enter the `t` action code to the left of the service class. When prompted, you provide the time range of 3:30 to 4:00 PM for each weekday of the previous week.

Entering the `t` action code on the Workload Manager Overview panel leads you to the Historical Trends for a Service Class Period panel pictured below.

For the designated time period of the previous week, the performance index (P/I) for period 1 was near 2, indicating actual response times of approximately twice the performance goal.
The historical trend display has quantified the complaint. Now, you can begin to investigate the cause of the problem.

**Example of GoTo pull-down**

Another way to display a historical trend panel is to use the GoTo pull-down. In this example, displayed by entering g, eleven categories are available for navigation.

If you choose selection 7 from this menu, OMEGAMON II displays historical trend information for system performance.
**Example of Index pull-down**

A third way to display a historical trend panel is to use the Index pull-down. In this example, displayed by entering fast path it, three categories of historical trend panels are available — workloads, I/O, and system.

When you select a category from this menu, OMEGAMON II displays another menu containing a list of the historical trend panels for that category.
Introduction

The flexibility of OMEGAMON II allows you to extract historical data and then manipulate that data to create customized statistical or graphic reports that meet the needs of your organization. The methods available to you are:

- the OBTAIN command, which enables you to export data to other applications
- the interface to SAS®, which produces graphical reports from the data you have exported.

Chapter Contents

Creating a Historical Output File .................................................. 348
Producing SAS Graphical Reports .................................................. 359
Creating a Historical Output File

Introduction

The OBTAIN command exports data from an EPILOG datastore to a sequential output file where it can be used by mainframe and PC application packages. This section shows how to create your own historical output file using the OBTAIN command.

In this section

In this section we cover the following topics.

- “Overview of how to create your own historical output file” on page 348
- “Preparing to Use the Sample Job or CLIST” on page 348
- “Defining the Data to be Exported” on page 349
- “Interpreting EBCDIC Columnar Output Files: FORMAT(COL)” on page 352
- “Interpreting Binary Output Files: FORMAT(INT)” on page 353
- “Interpreting EBCDIC Columnar Output Files with Delimiters: FORMAT(PC)” on page 356

Overview of how to create your own historical output file

These are the steps in exporting data using the OBTAIN command.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Edit a sample job or CLIST, setting the system parameters for your site.</td>
</tr>
<tr>
<td>2</td>
<td>Use the OBTAIN and SET commands to describe the data that you want to export.</td>
</tr>
<tr>
<td>3</td>
<td>Place your OBTAIN and SET commands in a dataset.</td>
</tr>
<tr>
<td>4</td>
<td>Submit the job or execute the CLIST.</td>
</tr>
</tbody>
</table>

Preparing to Use the Sample Job or CLIST

To export data with the OBTAIN command, you can either submit a batch job or execute a CLIST.

Sample files to help you run the OBTAIN command

Two sample files are included on the product tape to help you run the OBTAIN command.

<table>
<thead>
<tr>
<th>IF you want to...</th>
<th>THEN look at the sample file in...</th>
</tr>
</thead>
<tbody>
<tr>
<td>submit a batch job</td>
<td>rhileu.RKANSAM(KEPOBT)</td>
</tr>
<tr>
<td>execute a CLIST</td>
<td>rhileu.RKANCLI(KEPOBTC)</td>
</tr>
</tbody>
</table>
Creating a Historical Output File

**Note:** *rhelev* refers to the high-level qualifier for the dataset. If you need to find out what the high-level qualifier is for your installation, check with the person who installed OMEGAMON II.

**Modifying the sample files**
The job or CLIST must be modified to meet your needs.

<table>
<thead>
<tr>
<th>IF you need to change the definition of...</th>
<th>THEN modify DD statement...</th>
</tr>
</thead>
<tbody>
<tr>
<td>the output file</td>
<td>RKM2OUT, as explained in “Defining the output file(s)” on page 349</td>
</tr>
<tr>
<td>the data to be exported</td>
<td>SYSIN, as explained in “Defining the Data to be Exported” on page 349. This is done with the OBTAIN command.</td>
</tr>
<tr>
<td>the datastore from which the data is to be extracted</td>
<td>RKANPAR, which defines the library that contains the KEPEDS dataset. KEPEDS lists the datastores that can serve as the source of your data.</td>
</tr>
</tbody>
</table>

**Defining the output file(s)**
The output file(s) for the data that OBTAIN exports are defined as follows.

<table>
<thead>
<tr>
<th>IF you want...</th>
<th>THEN define...</th>
</tr>
</thead>
<tbody>
<tr>
<td>your entire command stream to write to a single output file</td>
<td>an available dataset in the RKM2OUT DD statement. This is the default method. Alternatively, you may define a different dataset using a different DD statement. In this case the DD statement must correspond to the DD name in the OUTFILE keyword of the OBTAIN command.</td>
</tr>
<tr>
<td>to place data in multiple output files</td>
<td>several available datasets by defining several DD statements. Each DD statement must correspond to a DD name in the OUTFILE keyword of an OBTAIN command in the OBTAIN command stream. You may then use multiple OBTAIN commands in one command stream to write to multiple output files.</td>
</tr>
</tbody>
</table>

**Defining the Data to be Exported**
Each OBTAIN command extracts requested information from EPILOG datastores and writes that information to a sequential output file that contains variable length records.

**What data can you extract?**
You can extract data elements detailing channel, CPU, and DASD resource utilization, as well as (for example) WLM service class statistics, and I/O queuing information. For a complete
description of the available data elements, see the discussion in the data dictionary portion of

**What data can’t you extract?**

Under MVS/SP 5, the historical collector gathers WLM-based data when MVS is running in
goal mode, and IPS-based data when MVS is running in compatibility mode. You cannot
extract WLM-based data elements from a datastore that was filled while MVS was running in
compatibility mode.

**Example of the OBTAIN command**

This sample OBTAIN command will export average response time data from yesterday for all
service class periods along with the statistic (PERFINDX) that shows how these response times
compare to their corresponding performance goals.

```
OBTAIN RSCL FORMAT(COL) -
    ELEMENTS(CLASNAME,CLASDESC,PERIODNM,AVGRESP,PERFINDX) -
    OUTFILE(SEQDATA) APPEND YESTERDAY
```

The APPEND keyword ensures that this OBTAIN command will not overwrite anything
already in SEQDATA. For complete information on all the OBTAIN keywords and all element
names see the EPILOG for MVS Command Language Reference Manual.

**Defining global parameters using the SET command**

When planning to code a command stream with several similar OBTAIN commands, you
may want to consider using the SET command to establish global default parameters for the
entire job. For example, if you wanted to gather a variety of information, all in
FORMAT(COL), you could precede your first OBTAIN command with this command:

```
SET FORMAT(COL)
```

You could then code as many OBTAIN commands as you like in that job without having to
include FORMAT(COL) on any of them.

For complete information on all of the keywords used with SET, see the EPILOG for MVS

**Types of output formats**

The format of the output file is controlled by the FORMAT keyword on an OBTAIN or SET
command in the command file.

<table>
<thead>
<tr>
<th>IF you need output in...</th>
<th>THEN use this keyword...</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBCDIC columnar format for possible conversion to ASCII</td>
<td>FORMAT(COL)</td>
</tr>
</tbody>
</table>
Creating a Historical Output File

Pointing to the OBTAIN commands
The RKM2IN DD statement in the sample job defines where the OBTAIN commands are located.

If the OBTAIN commands immediately follow this DD statement in the sample job, use the following statement:

```
//RKM2IN DD *
```

If the OBTAIN commands are in a separate dataset, use the following statement:

```
//RKM2IN DD dataset
```

where `dataset` is the fully qualified name of the dataset where the OBTAIN commands are located.

Submitting your batch job or running your CLIST
How you submit your batch job or run your CLIST depends on your particular site. For guidance, look in the appropriate sample file. Because it is difficult to estimate the number of observation records to expect, we recommend that you use generous secondary space allocations for your output file.

Interpreting your output
As previously explained, the format of your output file is controlled by the FORMAT keyword. For help in interpreting your output, please refer to the material indicated in the following table.

<table>
<thead>
<tr>
<th>IF you need output in...</th>
<th>THEN use this keyword...</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBCDIC columnar format <em>delimiters</em> for possible conversion to ASCII</td>
<td>FORMAT(PC)</td>
</tr>
<tr>
<td>binary format</td>
<td>FORMAT(INT)</td>
</tr>
</tbody>
</table>

### IF you need output in...

<table>
<thead>
<tr>
<th>THEN see...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Interpreting EBCDIC Columnar Output Files: FORMAT(COL)” on page 352</td>
</tr>
<tr>
<td>“Interpreting Binary Output Files: FORMAT(INT)” on page 353</td>
</tr>
<tr>
<td>“Interpreting EBCDIC Columnar Output Files with Delimiters: FORMAT(PC)” on page 356</td>
</tr>
</tbody>
</table>
Creating a Historical Output File

Interpreting EBCDIC Columnar Output Files: FORMAT(COL)

Output files created with the FORMAT(COL) keyword contain EBCDIC output in columns. Generating your output in this format is useful if you’re using the output files from OBTAIN as input files to mainframe graphics or statistics packages, or to certain PC software packages.

FORMAT(COL) record types

The FORMAT(COL) output file contains two record types.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>What It Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute record</td>
<td>Description of the contents of the observation records</td>
</tr>
<tr>
<td>Observation record</td>
<td>Element values selected from the datastore</td>
</tr>
</tbody>
</table>

Attribute record structure for FORMAT(COL)

Your output file will contain one attribute record for each OBTAIN command in the command stream. The record consists of a list of descriptions, one description for each element you requested. Each description contains the fields shown in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>element name, in character representation, preceded by an asterisk (*)</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>offset (position of the field from the beginning of the record) in integer representation</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>length of the element in bytes, in integer representation</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>data type (N=numeric signed integer, C=character, H=hexadecimal, or P=packed decimal)</td>
</tr>
</tbody>
</table>

The asterisk acts as a delimiter, flagging the beginning of the description of the next data element.

Observation records for FORMAT(COL)

The number of observation records depends on what workload or resource keyword you’re using. A workload keyword like JOBNAME, for example, may generate several observation records (one for each wait reason) per collection interval. You can reduce the number of observation records with the COMBINE keyword, which acts to combine data across collection intervals. Other keywords that may affect the number of observation records are PLOTMIN, REPORTIF, and SELECTIF.

Each observation record contains a value field for each data element selected by the user, in the order listed on the attribute record.

Command sequence example using FORMAT(COL)
Creating a Historical Output File

The following command stream generates one attribute record and 16 observation records.

```
SET FORMAT(COL)
SET OUTFILE(RSCLOUT) REPLACE
SET STIME(9) ETIME(11) SDATE(5/8) EDATE(5/8)
OBT RSCL ELEMENTS(CLASNAME,PERIODNM,PERFINDX,AVGRESP) -
   RIF(CLASNAME = TSO*)
END
```

The SET commands in this example specify the default format, output file, and reporting interval of all output files created by subsequent OBTAIN commands in this job.

**Output file example using FORMAT(COL)**

The following output file was generated by the previous sample command sequence.

<table>
<thead>
<tr>
<th>SDATE</th>
<th>STIME</th>
<th>CLASNAME</th>
<th>PERIODNM</th>
<th>PERFINDX</th>
<th>AVGRESP</th>
</tr>
</thead>
<tbody>
<tr>
<td>09412809:00:44</td>
<td>TSO</td>
<td>1</td>
<td>.500</td>
<td>475.4</td>
<td></td>
</tr>
<tr>
<td>09412809:00:44</td>
<td>TSOUSERS</td>
<td>1</td>
<td>4.246</td>
<td>424.6</td>
<td></td>
</tr>
<tr>
<td>09412809:15:23</td>
<td>TSO</td>
<td>1</td>
<td>.500</td>
<td>1508.1</td>
<td></td>
</tr>
<tr>
<td>09412809:15:23</td>
<td>TSOUSERS</td>
<td>1</td>
<td>4.050</td>
<td>405.0</td>
<td></td>
</tr>
<tr>
<td>09412809:30:38</td>
<td>TSO</td>
<td>1</td>
<td>.500</td>
<td>176.8</td>
<td></td>
</tr>
<tr>
<td>09412809:30:38</td>
<td>TSOUSERS</td>
<td>1</td>
<td>4.617</td>
<td>461.7</td>
<td></td>
</tr>
<tr>
<td>09412809:45:02</td>
<td>TSO</td>
<td>1</td>
<td>.500</td>
<td>695.1</td>
<td></td>
</tr>
<tr>
<td>09412809:45:02</td>
<td>TSOUSERS</td>
<td>1</td>
<td>4.419</td>
<td>441.9</td>
<td></td>
</tr>
<tr>
<td>09412810:00:07</td>
<td>TSO</td>
<td>1</td>
<td>.500</td>
<td>841.1</td>
<td></td>
</tr>
<tr>
<td>09412810:00:07</td>
<td>TSOUSERS</td>
<td>1</td>
<td>5.591</td>
<td>559.1</td>
<td></td>
</tr>
<tr>
<td>09412810:15:01</td>
<td>TSO</td>
<td>1</td>
<td>1.100</td>
<td>2637.2</td>
<td></td>
</tr>
<tr>
<td>09412810:15:01</td>
<td>TSOUSERS</td>
<td>1</td>
<td>3.665</td>
<td>366.5</td>
<td></td>
</tr>
<tr>
<td>09412810:30:00</td>
<td>TSO</td>
<td>1</td>
<td>.500</td>
<td>390.6</td>
<td></td>
</tr>
<tr>
<td>09412810:30:00</td>
<td>TSOUSERS</td>
<td>1</td>
<td>4.312</td>
<td>431.2</td>
<td></td>
</tr>
<tr>
<td>09412810:45:00</td>
<td>TSO</td>
<td>1</td>
<td>.700</td>
<td>562.0</td>
<td></td>
</tr>
<tr>
<td>09412810:45:00</td>
<td>TSOUSERS</td>
<td>1</td>
<td>3.573</td>
<td>357.3</td>
<td></td>
</tr>
</tbody>
</table>

The output file pictured above will be written to the dataset referenced by the OUTFILE keyword and defined on the corresponding RSCLOUT DD statement. The records displayed in the figure are in the same readable format as they would appear if you listed the file at a terminal or printer.

The number of observation records produced in this run is 16 because the default sampling interval is 15 minutes, the requested reporting interval is 2 hours (9 to 11), and the number of service class periods beginning with TSO is 2.

**Interpreting Binary Output Files: FORMAT(INT)**

Output files created with the FORMAT(INT) keyword contain binary data. Generating your output in this format is useful if you are writing your own programs to read the output file.

If you list the contents of an OBTAIN output file generated with FORMAT(INT), much of it is non-readable, binary data.

**FORMAT(INT) record types**
Creating a Historical Output File

The FORMAT(INT) output file contains the following record types.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>What It Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment record</td>
<td>Text of the OBTAIN commands in the command stream</td>
</tr>
<tr>
<td>Attribute record</td>
<td>Description of the contents of the observation records</td>
</tr>
<tr>
<td>Observation record</td>
<td>Element values selected from the datastore</td>
</tr>
</tbody>
</table>

**Comment record structure for FORMAT(INT)**
Your output file will contain one or more comment records for each OBTAIN command in the command stream.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>Julian date on which the data was extracted. Shown in packed decimal format.</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>time stamp showing the time of day that the data was extracted.</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>record-type indicator of *, indicating that this is a comment record.</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>reserved field</td>
</tr>
<tr>
<td>11</td>
<td>variable</td>
<td>text of the OBTAIN command. The first record starts with SDATE and STIME, followed by elements listed in the order you requested them. This field also contains the current settings of any valid keywords that are used for data filtering (such as DATE, TIME, BAND, RANGE, RIF, or SIF). For complete information on all of the keywords, refer to the EPILLOG for MVS Command Language Reference Manual.</td>
</tr>
</tbody>
</table>

**Attribute record structure for FORMAT(INT)**
Your output file will contain one attribute record for each OBTAIN command in the command stream. The record contains the fields shown in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>Julian date on which the data was extracted, shown in packed decimal representation.</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>time stamp showing the time of day that the data was extracted.</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>record-type indicator of A, indicating that this is an attribute record.</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>reserved field</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>the number of elements being OBTAINed (that is, the number of value fields that will appear in each observation record), shown in integer representation.</td>
</tr>
</tbody>
</table>
Creating a Historical Output File

As explained in the last row of the previous table, the element definition section of the attribute record starts in column 13 and includes 15 bytes of explanatory data for each data element requested. These 15 bytes are divided into the fields shown in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>15 per data element</td>
<td>the element definition of each element you requested, shown in character representation. The requested data elements are listed in the order in which you requested them.</td>
</tr>
</tbody>
</table>

As explained in the last row of the previous table, the element definition section of the attribute record starts in column 13 and includes 15 bytes of explanatory data for each data element requested. These 15 bytes are divided into the fields shown in the following table.

<table>
<thead>
<tr>
<th>Length</th>
<th>Field Description</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>element name</td>
<td>character</td>
</tr>
<tr>
<td>1</td>
<td>data type (I=numeric signed integer, U=unsigned integer, P=packed decimal, C=character, D=floating-point, X=hexadecimal, and T=time)</td>
<td>character</td>
</tr>
<tr>
<td>1</td>
<td>element length, in bytes</td>
<td>integer</td>
</tr>
<tr>
<td>1</td>
<td>decimal point displacement (number of positions to the right of the decimal; for example, CPU seconds may be shown to the nearest tenth of a second for a decimal point displacement of 1)</td>
<td>integer</td>
</tr>
<tr>
<td>4</td>
<td>field offset (location of the field from the beginning of the observation record, in bytes)</td>
<td>integer</td>
</tr>
</tbody>
</table>

The number of observation records depends on what workload or resource keyword you’re using. A workload keyword like JOBNAME, for example, may generate several observation records (one for each wait reason) per collection interval. You can reduce the number of observation records with the COMBINE keyword, which acts to combine data across collection intervals. Other keywords that may affect the number of observation records are PLOTMIN, REPORTIF, and SELECTIF.

Every observation record contains the fields shown in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>Julian date on which the data was extracted, shown in packed decimal representation.</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>time stamp showing the time of day that the data was extracted.</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>record-type indicator of O, indicating that this is an observation record.</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>reserved field</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>the field SDATE, which shows the Julian date on which the data was recorded. Shown in packed decimal representation.</td>
</tr>
</tbody>
</table>
Creating a Historical Output File

Command sequence example using FORMAT(INT)
The following sample command sequence generated three comment records, one attribute record, and many observation records.

```
SET FORMAT(INT)
OBT RSCL STIME(9) ETIME(11) SDATE(5/8) EDATE(5/8) -
   ELEMENTS(CLASNAME, PERIODNM, PERFINDX)
END
```

The SET command in this example specifies the default format of all output files created by subsequent OBTAIN commands in this job.

The output file for FORMAT(INT)
If you were to list the contents of an OBTAIN output file generated with FORMAT(INT), much of the data would be non-readable, binary data. It is for this reason that we do not provide a display of the output file generated with the sample FORMAT(INT) command stream.

Interpreting EBCDIC Columnar Output Files with Delimiters: FORMAT(PC)
Output files created with the FORMAT(PC) keyword contain EBCDIC output with delimiters. Generating your output in this format is useful if you are using the output files from OBTAIN as input files to a PC software package that requires delimiter characters.

FORMAT(PC) record types
The FORMAT(PC) output file contains two record types.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>What It Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute record</td>
<td>Description of the contents of the observation records</td>
</tr>
<tr>
<td>Observation record</td>
<td>Element values selected from the datastore</td>
</tr>
</tbody>
</table>

Attribute record example for FORMAT(PC)
Your output file will contain one attribute record for each OBTAIN command in the command stream. An example of an attribute record is shown below.

```
```

The record contains the element names that have been selected, delimited by double quotes.
Observation record structure for FORMAT(PC)

The number of observation records depends on what workload or resource keyword you’re using. A workload keyword like JOBNAME, for example, may generate several observation records (one for each wait reason) per collection interval. You can reduce the number of observation records with the COMBINE keyword, which acts to combine data across collection intervals. Other keywords that may affect the number of observation records are PLOTMIN, REPORTIF, and SELECTIF.

Every observation record contains the fields shown in the following table.

<table>
<thead>
<tr>
<th>Starting Column</th>
<th>Length</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>the field SDATE, which shows the Julian data on which the data was recorded. Shown in packed decimal format.</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>the field STIME, which shows the start time of the interval for which the data was recorded.</td>
</tr>
<tr>
<td>9</td>
<td>variable</td>
<td>value field for each element you selected, in the order listed in the attribute record. If the value is in character, time, or hexadecimal format, it will be enclosed in double quotes.</td>
</tr>
</tbody>
</table>

Command sequence example using FORMAT(PC)

The following command generates one attribute record and 16 observation records.

```
SET FORMAT(PC)
OBT RSCL STIME(9) ETIME(11) SDATE(5/8) EDATE(5/8) -
   ELEMENTS(CLASNAME, PERIODNM, PERFINDX, AVGRESP) -
   RIF(CLASNAME = TSO*)
END
```

The SET command in this example specifies the default format of all output files created by subsequent OBTAIN commands in this job.

Output file example using FORMAT(PC)

This output file is generated by the previous sample command sequence.
The records displayed in the figure are in the same readable format as they would appear if you listed the file at a terminal or printer.

Since the default RMF interval is 15 minutes, the requested reporting interval is two hours, and the number of service class periods starting with TSO is 2, 16 observation records are produced.

<table>
<thead>
<tr>
<th>SDATE</th>
<th>STIME</th>
<th>CLASNAME</th>
<th>PERIODNM</th>
<th>PERFINDX</th>
<th>AVGRESP</th>
</tr>
</thead>
<tbody>
<tr>
<td>094128</td>
<td>09:00:44</td>
<td>TSO</td>
<td>&quot;1&quot;</td>
<td>.500</td>
<td>475.4</td>
</tr>
<tr>
<td>094128</td>
<td>09:00:44</td>
<td>TSOUSERS</td>
<td>&quot;1&quot;</td>
<td>4.246</td>
<td>424.6</td>
</tr>
<tr>
<td>094128</td>
<td>09:15:23</td>
<td>TSO</td>
<td>&quot;1&quot;</td>
<td>.500</td>
<td>1508.1</td>
</tr>
<tr>
<td>094128</td>
<td>09:15:23</td>
<td>TSOUSERS</td>
<td>&quot;1&quot;</td>
<td>4.050</td>
<td>405.0</td>
</tr>
<tr>
<td>094128</td>
<td>09:30:38</td>
<td>TSO</td>
<td>&quot;1&quot;</td>
<td>.500</td>
<td>176.8</td>
</tr>
<tr>
<td>094128</td>
<td>09:30:38</td>
<td>TSOUSERS</td>
<td>&quot;1&quot;</td>
<td>4.617</td>
<td>461.7</td>
</tr>
<tr>
<td>094128</td>
<td>09:45:02</td>
<td>TSO</td>
<td>&quot;1&quot;</td>
<td>.500</td>
<td>695.1</td>
</tr>
<tr>
<td>094128</td>
<td>09:45:02</td>
<td>TSOUSERS</td>
<td>&quot;1&quot;</td>
<td>4.419</td>
<td>441.9</td>
</tr>
<tr>
<td>094128</td>
<td>10:00:07</td>
<td>TSO</td>
<td>&quot;1&quot;</td>
<td>.500</td>
<td>841.1</td>
</tr>
<tr>
<td>094128</td>
<td>10:00:07</td>
<td>TSOUSERS</td>
<td>&quot;1&quot;</td>
<td>5.591</td>
<td>559.1</td>
</tr>
<tr>
<td>094128</td>
<td>10:15:01</td>
<td>TSO</td>
<td>&quot;1&quot;</td>
<td>1.100</td>
<td>2637.2</td>
</tr>
<tr>
<td>094128</td>
<td>10:15:01</td>
<td>TSOUSERS</td>
<td>&quot;1&quot;</td>
<td>3.665</td>
<td>366.5</td>
</tr>
<tr>
<td>094128</td>
<td>10:30:00</td>
<td>TSO</td>
<td>&quot;1&quot;</td>
<td>.500</td>
<td>390.6</td>
</tr>
<tr>
<td>094128</td>
<td>10:30:00</td>
<td>TSOUSERS</td>
<td>&quot;1&quot;</td>
<td>4.312</td>
<td>431.2</td>
</tr>
<tr>
<td>094128</td>
<td>10:45:00</td>
<td>TSO</td>
<td>&quot;1&quot;</td>
<td>.700</td>
<td>562.0</td>
</tr>
<tr>
<td>094128</td>
<td>10:45:00</td>
<td>TSOUSERS</td>
<td>&quot;1&quot;</td>
<td>3.573</td>
<td>357.3</td>
</tr>
</tbody>
</table>
Producing SAS Graphical Reports

Introduction

OMEGAMON II provides an interface to the SAS graphics component. SAS converts technically complex data about system performance and resource utilization into concise, easy-to-read graphs that show you trends at a glance.

You can use these graphs to resolve performance problems, identify present and future resource needs, and ease management reporting.

*Note:* For detailed information on using the SAS program, see your SAS manual.

Purpose of the SAS interface

The SAS interface extracts resource, degradation, and profile data from the historical datastores and the Profile datastore and passes it to SAS for statistical reduction and graphic presentation.

This data can then be formatted into a variety of reports using a SAS program or added to an existing SAS performance database to enhance your current performance analysis system.

SAS interface requirements

In order to use the SAS interface, you need:

- SAS version 6.07 or 6.08
- a SAS-supported graphical device (or you will have to run the non-graphic versions of the report programs)
- a historical datastore with enough data to generate the graphs

In this section

In this section we cover the following topics:

- “Getting Started” on page 359
- “Understanding the Starter Kit” on page 361
- “Exporting Data to SAS” on page 363
- “Running the Graphics Replay Procedure” on page 366
- “Developing Your Own SAS Reports” on page 368

Getting Started

To use SAS, you must know how to supply parameters to the SAS programs, run the programs, and display the results.

SAS program parameters

Each SAS program in the thileu.TKANSAM dataset produces a separate report. Before running the program of your choice, you must modify certain parameters by setting user
overrides and providing data selection commands in the data extraction procedure identified by the &USERPROC variable. &USERPROC resolves correctly, based on the version of SAS that is running at your site.

**Running SAS programs**

You can run SAS programs online using the sample CLIST, or in batch mode using the sample batch job.

**Displaying results**

The sample CLIST and batch job allocate and free a temporary SAS file to hold the report output of the SAS program. The report is deleted after the program is run.

If you wish to save the report output for later use, you must allocate a permanent SAS graphics output file (also known as the graphics replay file) and define it to the CLIST or batch job. You can then use the graphics replay procedure to re-display the contents of the file online or print out the graphs to another device, such as a color plotter.

**Sample jobstream to allocate graphics replay file**

The sample jobstream for allocating a graphics replay file is provided in rhileu.RKANSAM(KEPGALLC) and is shown below.

```verbatim
// ... JOB CARD ... 
//*  THIS JOB ALLOCATES THE 'GRAFBASE' FILE USED BY KEPPMTSO and 
//*  KEPPMJCL.
//*  TO RUN THIS JOB:
//*  1. ADD A JOB CARD
//*  2. CHANGE THE 'SPACE' PARAMETER TO REFLECT YOUR SPACE REQMTS
//*  3. CHANGE 'VVVVVV' TO THE VOLSER YOU WISH TO ALLOC ON
//*  4. CHANGE 'ZZZZZZZZ' TO THE HIGH LEVEL QUALIFIER OF THE DATASET
//*    TO BE USED AS GRAFBASE
//*
//ALLOC    EXEC PGM=IEFBR14
//RKEPOUTG DD SPACE=(UUU,(P,S)), ENTER UNIT OF ALLOC, PRIM, SEC 
//        UNIT=SYSDA, 
//        VOL=SER=VVVVVV, 
//        DISP=(,CATLG,DELETE), 
//        DSN=ZZZZZZZZ.PRO.GRAFBASE, 
//        DCB=(DSORG=DA,RECFM=U,LRECL=6156,BLksz=6160)
```

**Allocating a graphics replay file**
Perform the following steps to complete the procedure for allocating a graphics replay file.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | Follow the comments provided in the sample jobstream.  
      | **Recommendations**  
      | ▪ Specify a space parameter of 5 cylinders to start.  
      | ▪ Specify on the **ZZZZZZZZ** parameter your TSO user ID or another high level qualifier that you will later define to the CLIST or batch job.  
| 2    | Run the job to allocate the file.  

Once you have allocated a graphics replay file, refer to it with the GRAPH parameter of the sample report.

Each time a SAS report is generated with the OMEGAMON II starter kit, the graphics replay file is overwritten with the output of the new program. Therefore, if you want to save the graphic output, you must allocate still another file and copy the current contents of the graphics replay file into it.

**Understanding the Starter Kit**

OMEGAMON II provides you with a starter kit to help you produce SAS reports. Each sample report included in the starter kit provides you with a picture of how various workloads or system resources are being utilized and which ones are constraining system performance.

**Types of SAS reports**

The model SAS reports in *thilev.TKANSAM* are grouped into two categories, as shown in the following table.

<table>
<thead>
<tr>
<th>Report Category</th>
<th>Member Names Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>KEPR</td>
</tr>
<tr>
<td>Batch Job and Programs</td>
<td>KEPBJ and KEPBP</td>
</tr>
</tbody>
</table>

Each report has a graphic and/or character graphic (SYSOUT) format. The member names in *thilev.TKANSAM* for the character graphic version have the usual KEP prefix followed by L (with some exceptions as noted). Example:

**KEPRLCBH**

The tables on the next few pages describe each SAS report contained in the starter kit. Reports are grouped by type and listed with their member names in *thilev.TKANSAM*. The first member name is used to produce the report in graphic form. The second member name is used for character graphic output.
## SAS resource reports

<table>
<thead>
<tr>
<th>SAS Report Name and PDS Member Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Channel Utilization Report</strong></td>
<td>Plots the average percent utilization per hour for selected Channel Path IDs over a specified time period. By specifying all channels, you produce a separate graph for each channel.</td>
</tr>
<tr>
<td><code>thilev.TKANSAM(KEPRCHBH)</code> or (KEPRLCBH)</td>
<td></td>
</tr>
<tr>
<td><strong>Channel Peak Utilization Report</strong></td>
<td>Plots the peak busy hour of the day for selected Channel Path IDs for each day in the specified time period. Each bar on the SAS chart represents the peak hour of a particular day, its height representing the percent utilization during that hour. By specifying all channels, you produce a separate graph for each channel.</td>
</tr>
<tr>
<td><code>thilev.TKANSAM(KEPRCHBP)</code> or (KEPRLCBP)</td>
<td></td>
</tr>
<tr>
<td><strong>Demand Paging Activity Report</strong></td>
<td>Shows the average demand paging activity for each hour of the day. Each point on the two-dimensional SAS graph represents the average demand paging rate for that hour. Separate plots are shown for page-in and page-out rates.</td>
</tr>
<tr>
<td><code>thilev.TKANSAM(KEPRPAGP)</code> or (KEPRLPAP)</td>
<td></td>
</tr>
<tr>
<td><strong>Swap Paging Activity Report</strong></td>
<td>Shows average swap paging activity for each hour of the day. Each point on the two-dimensional SAS graph represents the average swap paging rate for that hour. Separate plots are shown for page-in and page-out rates.</td>
</tr>
<tr>
<td><code>thilev.TKANSAM(KEPRPAGS)</code> or (KEPRLPAS)</td>
<td></td>
</tr>
<tr>
<td><strong>DASD Utilization Report</strong></td>
<td>Lists DASD devices in decreasing order of sustained peak utilization, which is determined as follows:</td>
</tr>
</tbody>
</table>
| `thilev.TKANSAM(KEPRDASD)` Character graphic output to printer. | 1. Data is extracted for the DASD devices during the time period you specify in the EXTRACT command, combined at 1 hour intervals.
2. For each device, an average hourly busy is calculated for each hour of the day over the days in the time period.
3. The program selects the averaged hour that had the highest percent busy value for each device. This is the sustained peak utilization.
4. The program sorts the devices in descending order of percent busy values. |
SAS batch job and program reports

<table>
<thead>
<tr>
<th>SAS Report Name and PDS Member Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Batch Job Comparison Report</strong>&lt;br&gt;thilev.TKANSAM(KEPBJOBC)&lt;br&gt;or (KEPBLJBC)</td>
<td>Compares a profile of the last run of a batch job with a profile of an average run of that job over a specified time period. Each profile consists of a breakdown of the main summary wait categories that contributed to the run time of the job. The output of the report is a bar graph, and the bands in the bars represent summary wait categories. This program uses only information from the EPILOG datastore.</td>
</tr>
<tr>
<td><strong>Batch Job Trending and Forecasting Report</strong>&lt;br&gt;thilev.TKANSAM(KEPBJOBT)&lt;br&gt;or (KEPBLJBT)</td>
<td>Examines the major summary wait categories that affect the run time for a given batch job. Generates two graphs:&lt;br&gt;  - a vertical bar chart that shows how the average run time for each run of the job was broken down by summary wait categories&lt;br&gt;  - a smooth trend of the run time over the selected time period, with different colored bands in the graph showing how each summary wait category contributed to overall run time&lt;br&gt;SAS ETS software creates a third graph, which forecasts the job for its next few executions.</td>
</tr>
<tr>
<td><strong>Batch Program Resource Consumption Report</strong>&lt;br&gt;thilev.TKANSAM(KEPBPGML)&lt;br&gt;Character graphic version to printer.</td>
<td>Describes the resources consumed by your batch programs. The report contains three parts:&lt;br&gt;1. Part 1 lists batch programs that ran during the specified time period by frequency of execution. Programs are listed in decreasing order, with the most frequently run program listed first.&lt;br&gt;2. Part 2 lists programs in decreasing order by CPU consumption. The program that used the most CPU time is listed first.&lt;br&gt;3. Part 3 lists programs in decreasing order by active I/O time. The program that used the most active I/O time is listed first.</td>
</tr>
</tbody>
</table>

Exporting Data to SAS

Once you know how SAS generates reports and understand the function of each report, you are ready to produce reports of your own.

Modeling reports

The easiest way to produce SAS reports is to model them after sample reports included in the SAS starter kit. You choose a report, modify the user overrides, and provide the required keywords.
Overview of producing reports

You produce reports from the starter kit with the following procedure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Locate the member in thilev.TKANSAM that contains the SAS program for the report you wish to produce. &lt;br&gt;Report descriptions and members are found under “Understanding the Starter Kit” on page 361.</td>
</tr>
<tr>
<td>2</td>
<td>Copy this member into another member so that the original member remains unchanged.</td>
</tr>
<tr>
<td>3</td>
<td>Using an editor, modify the user overrides and data extraction procedure &amp;USERPROC contained in the copy as directed in the comments provided in the member.</td>
</tr>
<tr>
<td>4</td>
<td>Run the SAS program, including in the CLIST or JCL the PROG(cccccc) parameter, where ccccccc is the name of the copy.</td>
</tr>
<tr>
<td>5</td>
<td>Run the graphics replay procedure (optional).</td>
</tr>
</tbody>
</table>

**User overrides**

The complete list of user overrides is described below. The overrides used by individual reports are identified in the thilev.TKANSAM member for that report.

**BATNAME**: The name of the batch job.

**CHANNEL**: Specifies the channel number you would like to report on for the channel utilization and I/O rate reports. To produce graphs for all channels in your system, enter 999.

**DDNAME**: Defines the ddname of the output SAS data library specified in the CLIST or batch job that runs the report.

**DEVICE**: Defines the name of the output graphic device driver. For example, if you are directing the output to a 3279 terminal, enter IBM32793.

**FORECAST**: If you have SAS ETS software installed, you can run additional forecasting sections for the KEPBJOBT report by setting this override to YES. Set to NO if you do not have SAS ETS.

**KEPCOLOR**: Sets the color formatting for the output device. If the output device is a color graphics terminal, enter CRT. If you are using another output device to produce hardcopy, enter HARD. (This override is actually a macro call. You can supply your own version of the KEPCOLOR macro to match your hardware configuration.)

**LEAD**: If you are running the forecasting section of the KEPBJOBT report, LEAD allows you to set the number of job runs to be forecasted. Generally, this should not exceed 10% of the number of runs in the report.
Producing SAS Graphical Reports

Identifying the user overrides to modify
Each of the sample programs in the starter kit contains a set of user overrides that control SAS formatting, report titling, and data selection.

A sample set of user overrides is shown in the following figure.

```
+------------------------------------------------------------------+
|     USER OVERRIDES - CHANGE JJJJJJJJJJJ TO DESIRED JOBNAME         |
+------------------------------------------------------------------+
%LET BATNAME=JJJJJJJJJJ;  /* SELECT THE JOB TO BE CHARTED           */
%LET DEVICE=IBM32793 ;   /* SELECT GRAPHICS DEVICE DRIVER NAME     */
%LET TERMINAL=TERMINAL ; /* TERMINAL=SHOW GRAPHS IMMEDIATELY       */
/* NOTERMINAL=DON’T SHOW GRAPHS NOW                                */
%LET DDNAME=EPPRO ;      /* DDNAME OF &USERPROC OUTPUT             */
%KEPCOLOR (CRT )         /* HARD=HARD COPY, CRT=CRT SCREEN KEPCOLOR*/
```

The comments in the member tell you how to modify these overrides for this report.

Modifying the data extraction procedure
The data extraction procedure represented by &USERPROC is in every member of thileu.TKANSAM. A sample procedure is shown below.
Producing SAS Graphical Reports

As you see, the data extraction procedure represented by &USERPROC consists of four components:

1. The procedure &USERPROC statement, which includes the ddname parameter. This ddname is set to the SAS output file name defined in the user overrides. (&DDNAME does this automatically.)

2. The PARMCARDS line, which signals the beginning of the input cards for the procedure. This line must end with a semicolon.

3. The input cards for the procedure, which are one or more EXTRACT (EXT) or COMPEXT (CMX) commands that select the data from the EPILOG datastore and convert it into a format that is compatible with SAS processing.

   EPILOG commands cannot be written on the same line as the PARMCARDS statement, nor can they be terminated with a semicolon.


4. The terminating semicolon for the input cards must be written on the line following the last input card.

Running the program

There are two ways to run the program without keeping the report output.

<table>
<thead>
<tr>
<th>IF you want to run the program...</th>
<th>THEN edit...</th>
</tr>
</thead>
<tbody>
<tr>
<td>interactively</td>
<td>the sample CLIST provided in rhilu.RKANCLI(KPMTSO6).</td>
</tr>
<tr>
<td>in batch mode</td>
<td>the sample JCL provided in rhilu.RKANSAM(KPJMCL6).</td>
</tr>
</tbody>
</table>

Running the Graphics Replay Procedure

The graphics replay procedure re-displays SAS report output saved in the graphics replay file by directing the contents of the file to a terminal or to another device, such as a color plotter.

Before you begin

The graphics replay procedure will only work if you have already allocated a graphics replay file and run the CLIST or batch job to populate the output file. For complete information on allocating a graphics replay file, see “Getting Started” on page 359.
The graphics replay program
The SAS program to run the graphics replay procedure is contained in `thileu.TKANSAM(KEPGREPL)` and is shown below.

```sas
%INC RKANSAM(KEPPROVE) / SOURCE2 ;
%MACROS
%INCLUDE RKANSAM(KEPVERSI) / SOURCE2 ;
%INCLUDE RKANSAM(KEPMREPL) / SOURCE2 ;
%LET DEVICE=IBM32793; /* SPECIFY THE REPLAY DEVICE DRIVER */
%LET TERMINAL=TERMINAL; /* TERMINAL=SHOW GRAPHS IMMEDIATELY */
%LET DDNAME=GRAPH; /* DDNAME OF STORED GRAPHIC DISPLAY FILE */
%LET DSNAME=JOBDATA; /* SAS DATA SET NAME OF GRAPH FILE */
%KEPVERSI /* DETERMINE SAS VERSION BEING EXECUTED */
RUN;
GOPTIONS DEVICE=&DEVICE &TERMINAL ;
OPTIONS &TEXT82 DQUOTE ;
%KEPMREPL /* RUN THE CORRECT KEPGREPL FOR SAS VERSION*/
ENDSAS;
```

Graphics replay program user overrides
The graphics replay program contains four user overrides.

**DEVELOP**

The output device identifier. This should correspond to the identifier entered as the DEVICE override in the SAS program. If different identifiers are entered, incompatibilities may cause such problems as visual distortion or abnormal session termination.

**TERMINAL**

If a terminal was specified as the output device identifier, enter TERMINAL in this override. If another identifier was specified, enter NOTERMINAL as the override.

**DDNAME**

The ddname of the graphics replay program, as defined by the ddname of the JCL used to run the procedure. If you are running reports from the starter kit, you should leave the DDNAME override set to GRAPH.

**DSNAME**

The second level qualifier of the dataset name used during graph generation. In the starter kit programs, this is set to JOBDATA.

### Running the graphics replay procedure

Perform the following steps to run the graphics replay procedure:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Copy $thilev$.TKANSAM(KEPGREPL) into another member, say MYREPLAY, of the same dataset so that the original remains unchanged.</td>
</tr>
<tr>
<td>2</td>
<td>Modify the user overrides according to your needs.</td>
</tr>
<tr>
<td>3</td>
<td>Replay the output in one of the following ways:</td>
</tr>
<tr>
<td></td>
<td>To replay the graphs on a terminal, enter the following command:</td>
</tr>
<tr>
<td></td>
<td><code>%KPMTSO6 PROG(myreplay)</code></td>
</tr>
<tr>
<td></td>
<td>Result: A menu containing each graph generated by the report is displayed. Select the graphs you want to see by placing an S next to the graph name on the menu.</td>
</tr>
<tr>
<td></td>
<td>To replay the graphs to a hardcopy device, run the sample batch job with the name of your copy of KEPRCHBH as the PROG parameter.</td>
</tr>
</tbody>
</table>

### Developing Your Own SAS Reports

Now that you know how to produce SAS reports, it is time to develop two examples.

#### Developing a resource report

Suppose you want to produce a Channel Utilization Report and direct the output to your terminal:

- Locate the PDS member name $thilev$.TKANSAM(KEPRCHBH) for the report in “Understanding the Starter Kit” on page 361 and copy its contents to another member, say MYCOPY.

- Edit MYCOPY and modify the six user overrides and EXTRACT keywords contained in the data extraction procedure represented by &USERPROC.
To run the report, enter the following command:

```
%KPMTSO6 PROG(mycopy)
```

The first graph is displayed. If you specified CHANNEL=999 in the user overrides, a graph is produced for each channel in your system. Press F3 or F15 to display the next graph.

### Developing a Batch Job Trending and Forecasting Report

Now you want to produce a Batch Job Trending and Forecasting Report. This time you want to run the report using the batch job and then use the graphics replay procedure under TSO to display it on the terminal.

The initial steps are the same as before, except that the original member is `thileu.TKANSAM(KEPBJOBT)`. The user overrides and the data extraction procedure 

```sas
&USERPROC are shown below.
```

```
/* WE WILL AVERAGE ONLY WEEK DAYS */
PROC &USERPROC DDNAME=&DDNAME ;
PARMCARDS ;
EXT RCHN SDATE(-30) DAY(WKDAY) CMB(1H)
;```
Submit the following job:

```plaintext
//JOB...
// EXEC KPMJCL6, PROG=mycopy
```

When the job finishes, copy `thileu.TKANSAM(KEPGREPL)` into another member of the same dataset, edit the user overrides, and execute the appropriate CLIST using your member name for `KEPGREPL` as the `PROG` parameter.

**SAS macros**

Nine macros were used to write the programs for the starter kit reports, and you may want to use or modify these macros if you write your own programs.

The following table provides a definition of each macro.

<table>
<thead>
<tr>
<th>Macro</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEPBATCHM</td>
<td>Used to condense and simplify data in the batch job trending report. The macro groups several related major wait reasons into summary wait categories.</td>
</tr>
<tr>
<td>KEPCOLLS, KEPCOLLC</td>
<td>When a starter-kit program generates more than one graph with SAS, KEPCOLLS starts the process of collecting them from a temporary dataset by writing the first graph to a permanent dataset (GRAPH.JOBDATA). KEPCOLLC continues the process by writing the subsequent graphs to the same permanent dataset.</td>
</tr>
<tr>
<td>KEPCOLOR</td>
<td>Sets eight global macro variables to define the color scheme on the output device. The version that is distributed with the product has an input variable that can be set to CRT or HARD. CRT sets the color scheme for an IBM 3279 terminal. HARD sets the color scheme for a color plotter. Modify the color settings for the HARD option to correspond with your hardcopy output device, if necessary.</td>
</tr>
<tr>
<td>Macro</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>KEPELGR</td>
<td>When a starter-kit program generates a new graph(s) with SAS, this macro deletes the old version(s) from the graphics replay file.</td>
</tr>
<tr>
<td>KEPMREPL</td>
<td>Conditionally executes the KEPGREPL procedure. It is used only in the starter-kit program KEPGREPL.</td>
</tr>
<tr>
<td>KEPEVERSI</td>
<td>Uses the SAS automatic variable &amp;SYSVER to determine which version of SAS is being executed. It then modifies release-dependent macro variables accordingly.</td>
</tr>
</tbody>
</table>
Introduction

This chapter presents the information you need to run a dedicated OMEGAMON II session when VTAM services are unavailable.

Note: When your system is operating in goal mode, VTAM is used to communicate with the Candle Management Server. When VTAM services are unavailable, the information provided by your OMEGAMON II system, particularly that related to the workload manager, will be diminished.

You will be using a locally-attached, non-SNA, 327n terminal, which is dedicated to a special OMEGAMON II address space that does not require VTAM telecommunications services. This special non-VTAM OMEGAMON II address space can run alone or while the regular OMEGAMON II address space is running.

Note: When running OMEGAMON II for MVS in dedicated mode, VTAM is required to connect to the Candle Management Server.

Where can you run OMEGAMON II from a dedicated (Non-VTAM) terminal?

The locally-attached, non-SNA, 327n terminal that is dedicated to the non-VTAM OMEGAMON II address space is typically located in the data center close to the operator console.

Limitations

You cannot zoom to OMEGAMON and EPILOG since the special OMEGAMON II address space does not use VTAM services.

Prerequisite

The instructions in this chapter assume that the non-VTAM OMEGAMON II address space has already been started. By default, the command to do this is:

S km2pdmod

Appendix Contents

Signing Onto and Off OMEGAMON II ................................................................. 374
Adding and Releasing Sessions ................................................................. 375
Changing the Polling Interval ................................................................. 376
Setting up and Using Automatic Sign On .................................................. 377
Signing Onto and Off OMEGAMON II

The instructions on this page describe how to sign onto and off the non-VTAM OMEGAMON II address space. The instructions must be entered at the locally-attached, non-SNA, 327n terminal that has been dedicated to the non-VTAM OMEGAMON II address space.

Signing on

To sign on, enter the instructions below at the device that is dedicated to the non-VTAM OMEGAMON II address space.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>After the non-VTAM OMEGAMON II address space is started, the Candle copyright screen appears. Press Enter to display the OMEGAMON II Sign On Panel.</td>
</tr>
<tr>
<td>2</td>
<td>On the OMEGAMON II Sign On Panel, enter your user ID and password, and press Enter. Result: The Establishing OMEGAMON II Environment panel appears, followed by the System Status panel.</td>
</tr>
</tbody>
</table>

Signing off

Perform these steps to sign off your OMEGAMON II session.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From any OMEGAMON II panel, press F3 until the Exit Confirmation panel appears.</td>
</tr>
<tr>
<td>2</td>
<td>Press F3, or enter x to return to the Sign On Panel.</td>
</tr>
</tbody>
</table>
Adding and Releasing Sessions

By default, when the non-VTAM OMEGAMON II address space is started, instructions in the km2pdmod started task assign one locally attached (dedicated) device to this OMEGAMON II address space.

You can dedicate as many devices as you wish to the non-VTAM OMEGAMON II address space, but you can run a maximum of three non-VTAM OMEGAMON II sessions concurrently. The instructions on this include two ways to add additional sessions and one way to release them.

Adding an additional session

You can dedicate an additional device to the non-VTAM OMEGAMON II address space from an MVS console, or you can arrange to have multiple devices dedicated each time the address space is started.

<table>
<thead>
<tr>
<th>IF you want to dedicate...</th>
<th>THEN enter this command, as directed...</th>
</tr>
</thead>
<tbody>
<tr>
<td>an additional device from the MVS console</td>
<td>F km2pdmod,DEDICATE UNIT-&lt;cUU&gt;</td>
</tr>
<tr>
<td>multiple devices at each start-up</td>
<td>DEDICATE UNIT-&lt;cUU&gt;</td>
</tr>
</tbody>
</table>

Releasing a session

You can release a device from being dedicated to the non-VTAM OMEGAMON II address space. To release a device, enter this command from an MVS console:

F km2pdmod,CLOSE UNIT-<cUU> | DEV-<nnnn>

where <cUU> is the dedicated device address, or <nnnn> is the 4-digit device number.
Changing the Polling Interval

What is the polling interval?
Since your OMEGAMON II session does not use VTAM services, your keyboard requests are not handled on “interrupt” basis. That is, a keyboard request is not sent to OMEGAMON II each time you press an action key (Enter, or a Function key).

Instead, OMEGAMON II periodically reads the terminal output buffer and then processes the request. A complete keyboard request consists of one action key and any number of characters. The reads are performed at a preset time interval called the polling interval.

By default, the polling interval is 1 second.

Changing the polling interval
You can reset the polling interval by issuing two MVS console commands, or by entering a DEDICATE statement in a member that is processed each time the non-VTAM OMEGAMON II address space is started.

<table>
<thead>
<tr>
<th>IF you want to reset the polling interval...</th>
<th>THEN enter the command(s), where directed...</th>
</tr>
</thead>
<tbody>
<tr>
<td>from the MVS console</td>
<td>* F km2pdm, CLOSE UNIT-cuu</td>
</tr>
<tr>
<td></td>
<td>* F km2pdm, DEDICATE UNIT-cuu</td>
</tr>
<tr>
<td>with a statement that is processed at each start-up</td>
<td>DEDICATE UNIT-cuu</td>
</tr>
<tr>
<td></td>
<td>in the KM2DEDS member of the rhilev.RKANCMD dataset. Then start or restart the km2pdm started task.</td>
</tr>
</tbody>
</table>

DEDICATE parameters
This table describes the parameters on the DEDICATE statements in both of the preceding procedures:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>The number of seconds for the polling interval. Typically, polling intervals do not exceed 5 seconds.</td>
</tr>
<tr>
<td>cuu</td>
<td>The device address in MVS/370.</td>
</tr>
<tr>
<td>nnnn</td>
<td>The device number in MVS/XA™.</td>
</tr>
<tr>
<td>KM2ENTDL</td>
<td>The default name of the initial dialog procedure.</td>
</tr>
<tr>
<td>NODELAY</td>
<td>Eliminates the need to press Enter during the sign-on process in order to advance to the Candle copyright screen.</td>
</tr>
</tbody>
</table>
Setting up and Using Automatic Sign On

The automatic sign-on feature eliminates the need to enter your user ID and password when you access the non-VTAM OMEGAMON II address space from a dedicated device.

After setting up for automatic sign-on, you will have to perform a normal sign-on once, as described previously in this appendix. You will not have to sign on to the dedicated session again until your password expires or the device number of the dedicated device you are using changes.

Setting up for automatic sign-on

You can set up for automatic sign on from an MVS console, or you can have automatic sign on set up each time the km2pdmod started task is started.

<table>
<thead>
<tr>
<th>IF you want to set up automatic sign on...</th>
<th>THEN enter this command, where directed...</th>
</tr>
</thead>
<tbody>
<tr>
<td>from the MVS console</td>
<td>F km2pdmod, DEDICATE UNIT-cuu</td>
</tr>
<tr>
<td></td>
<td>from the MVS console</td>
</tr>
<tr>
<td>with a statement that is processed at each start-up</td>
<td>DEDICATE UNIT-cuu</td>
</tr>
<tr>
<td></td>
<td>in the KM2DEDS member of the rhilev. RKANCMD dataset. Then start or restart the km2pdmod started task.</td>
</tr>
</tbody>
</table>

DEDICATE parameters

This table describes the parameters on the DEDICATE statements in both of the procedures above:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cuu</td>
<td>The device address in MVS/370</td>
</tr>
<tr>
<td>nnn</td>
<td>The device number in MVS/XA</td>
</tr>
<tr>
<td>KM2ENTDA</td>
<td>The name of the initial dialog procedure that supports automatic sign on.</td>
</tr>
<tr>
<td>NODELAY</td>
<td>Eliminates the need to press Enter during the sign-on process in order to advance to the Candle copyright screen.</td>
</tr>
</tbody>
</table>

Accessing an automatic sign-on session

When you start up the special OMEGAMON II address space, the Candle copyright screen appears. After 5 seconds, the Establishing OMEGAMON II Environment panel appears, followed by the System Status panel.

At this point you have access to the OMEGAMON II session.
Setting up and Using Automatic Sign On

Signing off

Perform these steps to sign off the automatic-sign-on session.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From any OMEGAMON II panel, press F3 until the Exit Confirmation panel appears.</td>
</tr>
<tr>
<td>2</td>
<td>Press F3, or enter x to return to the System Status Panel.</td>
</tr>
</tbody>
</table>

Why sign off?

Signing off an automatic sign-on session may appear to be a meaningless exercise, but accomplishes the following adjustments to the OMEGAMON II operating environment:

- Resets the security password to its default value.
- Saves any profile and user authority settings entered since the previous sign off.
Introduction

Candle Corporation is committed to producing top-quality software products and services. To assist you with making effective use of our products in your business environment, Candle is also committed to providing easy-to-use, responsive customer support.

Precision, speed, availability, predictability—these terms describe our products and Customer Support services.

Included in this Guide to Candle Customer Support is information about the following:

Base Maintenance Plan .............................................. 380
  – Telephone Support
  – eSupport
  – Description of Severity Levels
  – Service-level objectives
  – Recording and monitoring calls for quality purposes
  – Customer Support Escalations
  – Above and Beyond

Enhanced Support Services ........................................... 384
  – Assigned Support Center Representative (ASCR)
  – Maintenance Assessment Services (MAS)
  – Multi-Services Manager (MSM)

Customer Support Contact Information .............................. 385
  – Link to Worldwide Support Telephone and E-mail information
Base Maintenance Plan

Overview
Candle offers a comprehensive Base Maintenance Plan to ensure that you realize the greatest value possible from your Candle software investments. We have more than 200 technicians providing support worldwide, committed to being responsive and to providing expedient resolutions to support requests. Technicians are available worldwide at all times during the local business day. In the event of an after-hours or weekend emergency, our computerized call management and forwarding system will ensure that a technician responds to Severity One situations within one hour. For customers outside of North America, after-hours and weekend support is provided in English language only by Candle Customer Support technicians located in the United States.

Telephone support
Candle provides consistently reliable levels of service—thanks to our worldwide support network of dedicated experts trained for specific products and operating systems. You will always work with a professional who truly understands your problem.

We use an online interactive problem management system to log and track all customer-reported support requests. We give your support request immediate attention by routing the issue to the appropriate technical resource, regardless of geographic location.

Level 0 Support is where your call to Candle Customer Support is first handled. Your support request is recorded in our problem management system, then transferred to the appropriate Level 1 support team. We provide Level 0 manual interaction with our customers because we support more than 170 products. We feel our customers would prefer personal interaction to a complex VRU or IVR selection menu.

Level 1 Support is the service provided for initial support requests. Our Level 1 team offers problem determination assistance, problem analysis, problem resolutions, installation assistance, and preventative and corrective service information. They also provide product usage assistance.

Level 2 Support is engaged if Level 1 cannot provide a resolution to your problem. Our Level 2 technicians are equipped to analyze and reproduce errors or to determine that an error is not reproducible. Problems that cannot be resolved by Level 2 are escalated to Candle’s Level 3 R&D support team.

Level 3 Support is engaged if a problem is identified in Candle product code. At Level 3, efforts are made to provide error correction, circumvention or notification that a correction or circumvention is not available. Level 3 support provides available maintenance modifications and maintenance delivery to correct appropriate documentation or product code errors.
**eSupport**

In order to facilitate the support process, Candle also provides **eSupport**, an electronic full-service information and customer support facility, using the World Wide Web at [www.candle.com/support/](http://www.candle.com/support/). **eSupport** allows you to open a new service request and update existing service requests, as well as update information in your customer profile. New and updated service requests are queued to a support technician for immediate action. And we can respond to your request electronically or by telephone—it is your choice.

**eSupport** also contains a continually expanding knowledge base that customers can tap into at any time for self-service access to product and maintenance information.

The Candle Web Site and **eSupport** can be accessed 24 hours a day, 7 days a week by using your authorized Candle user ID and password.

**Description of Candle severity levels**

Responses to customer-reported product issues and usage questions are prioritized within Candle according to Severity Code assignment. Customers set their own Severity Levels when contacting a support center. This ensures that we respond according to your individual business requirements.

<table>
<thead>
<tr>
<th>Severity Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Severity 1</strong></td>
<td>A crisis affects your ability to conduct business, and no procedural workaround exists. The system or application may be down.</td>
</tr>
<tr>
<td><strong>Severity 2</strong></td>
<td>A high-impact problem indicates significant business effect to you. The program is usable but severely limited.</td>
</tr>
<tr>
<td><strong>Severity 3</strong></td>
<td>A moderate-impact problem involves partial, non-critical functionality loss or a reasonable workaround to the problem. A “fix” may be provided in a future release.</td>
</tr>
<tr>
<td><strong>Severity 4</strong></td>
<td>A low-impact problem is a “how-to” or an advisory question.</td>
</tr>
<tr>
<td><strong>Severity 5</strong></td>
<td>This is a request for software or documentation enhancement. Our business units review all requests for possible incorporation into a future release of the product.</td>
</tr>
</tbody>
</table>
Candle has established the following service-level objectives:

<table>
<thead>
<tr>
<th>Call Status</th>
<th>Severity 1 Goal</th>
<th>Severity 2 Goal</th>
<th>Severity 3 Goal</th>
<th>Severity 4 Goal</th>
<th>Severity 5 Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Call Time to Answer</td>
<td>90% within one minute</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level 1 Response (Normal Business Hours)</td>
<td>90% within 5 minutes</td>
<td>90% within one hour</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level 2 Response (Normal Business Hours)</td>
<td>Warm Transfer</td>
<td>90% within two hours</td>
<td>90% within eight hours</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scheduled follow-up (status update)</td>
<td>Hourly or as agreed</td>
<td>Daily or as agreed</td>
<td>Weekly or as agreed</td>
<td>Notification is made when an enhancement is incorporated into a generally available product.</td>
<td>Notification is made when a fix is incorporated into a generally available product.</td>
</tr>
</tbody>
</table>

The above information is for guideline purposes only. Candle does not guarantee or warrant the above service levels. This information is valid as of October 1999 and is subject to change without prior notice.

Recording and Monitoring Calls for Quality Purposes

Candle is committed to customer satisfaction. To ensure that our customers receive high levels of service, quality and professionalism, we'll monitor and possibly record incoming and outgoing Customer Support calls. The information gleaned from these calls will help us serve you better. If you prefer that your telephone call with Candle Customer Support in North America not be monitored or recorded, please advise the representative when you call us at (800) 328-1811 or (310) 535-3636.

Customer Support Escalations

Candle Customer Support is committed to achieving high satisfaction ratings from our customers. However, we realize that you may occasionally have support issues that need to be escalated to Candle management. In those instances, we offer the following simple escalation procedure:

If you experience dissatisfaction with Candle Customer Support at any time, please escalate your concern by calling the Candle support location closest to you. Ask to speak to a Customer Support manager. During standard business hours, a Customer Support manager will be available to talk with you or will return your call. If you elect to hold for a manager, you will be connected with someone as soon as possible. If you wish a return call, please tell the Candle representative coordinating your call when you will be available. After contacting you, the Customer Support manager will develop an action plan to resolve your issue. All escalations or complaints received about support issues are logged and tracked to ensure responsiveness and closure.
Above and Beyond

What differentiates Candle’s support services from our competitors? We go the extra mile by offering the following as part of our Base Maintenance Plan:

- Unlimited multi-language defect, installation and operations support
- eSupport using the World Wide Web
- Regularly scheduled product updates and maintenance provided at no additional charge
- Over 200 specialized technicians providing expert support for your Candle products
Enhanced Support Services

Overview

Our Base Maintenance Plan provides a high level of software support in a packaged offering. However, in addition to this plan, we have additional fee-based support services to meet unique customer needs.

The following are some examples of our added-value support services:

- **Assigned Support Center Representative Services (ASCR)**
  - An assigned focal point for managing support escalation needs
  - Proactive notification of available software fixes
  - Proactive notification of product version updates
  - Weekly conference calls with your ASCR to review active problem records
  - Monthly performance reviews of Candle Customer Support service levels
  - Optional on-site visits (extra charges may apply)

- **Maintenance Assessment Service (MAS)**
  - On-site assessment services
  - Advice about product maintenance and implementation
  - Training your staff to develop efficient and focused procedures to reduce overall cost of ownership of your Candle software products
  - Analysis of your Candle product environment: versions, updates, code correction history, incident history and product configurations
  - Reviews to ensure that purchased Candle products and solutions are used effectively

- **Multi-Services Manager (MSM)**
  Multi-Services Manager provides highly valued services to customers requiring on-site full time expertise to complement their technical resources.
  - Dedicated on-site Candle resource (6 months or one year) at your site to help ensure maximum use and effectiveness of your Candle products
  - Liaison for all Candle product support activities, coordination and assistance with implementation of all product updates and maintenance releases
  - Works with your staff to understand business needs and systems requirements
  - Possesses technical and systems management skills to enhance your staff’s knowledge and expertise
  - Other projects as defined in Statement of Work for MSM services
Customer Support Contact Information

**Link to Worldwide Support Telephone and E-mail information**

To contact Customer Support, the current list of telephone numbers and e-mail addresses can be found on the Candle Web site, [www.candle.com/support/](http://www.candle.com/support/).

Select **Support Contacts** from the list on the left of the page.
Symbols

! (exclamation) symbol 32
&O exceptions
  migrating to &OII 55
&USERPROC 359
() parentheses 36
+ (plus sign) 28
/AUPOFF command 133
.WAIT command 133
.AUPOFF command 311
.FGO command 170, 176, 179
  looping with 170
.LOG command 172
  POP 172
  PUSH 172
.LOGON 172
.LOGPOP 172
.LOGPUSH 172
Monitoring workloads, resources, and alerts 71
.OPTN command 175
.RTN command 175, 176, 178, 179
.SET command 133, 311
.SET INTERVAL command 132
.SGO command 173, 176, 179
.TSF 172
.TSF command 179
.TSF00 command 179
.VAR immediate command 171
.XLF 172
.XSUM command 175
/ (slash) 143, 148
/ATTACH command 130
/AUPOFF command 311
/BOTTOM command 161
/DEF HOLD INFO-line command 159
/DEF OFF INFO-line command 159
/DEF ON INFO-line command 159
/GIVE command 132
/LOG command 176
/REP INFO-line command 159, 161
/SAVE INFO-line command 161
/STOP INFO-line command 160
/TAKE command 132
/TOP command 161
/WAIT command 310
<> (brackets) 36
> (arrow) symbol 32
> (comment character) 161
>> (double arrow) symbol 32

Numerics

2105 284, 289
3880 284
3990 284

A
A action code 34, 68, 145, 250
access methods 187
accessing batch report controls 192
acting upon an object 23
  pressing Enter 23
  typing a mnemonic 23
  typing a slash (/) 23, 26, 34, 127, 305
action
  T (historical trends) 76
action bar
  Actions 22
  function keys 21
  GoTo 22
  Help 22
  Index 22
  location of 21
  Options 22
  panel body 21
  use of 23
  View 22
action code
  using 34
action codes 34
  A (analyze details) 250
  A (analyze problems) 34, 54, 68, 145
  B (Bottlenecks) 34
  B (bottlenecks) 68, 139, 146, 315
  H (Historical details) 338
  H (historical details) 34, 68, 70, 183, 250, 252
  L (control data collection frequency)
  L (control data collection frequency) 51, 249
  L (threshold persistence) 34
  S (show details) 34, 68, 69, 70, 72, 124, 250,
    251, 252, 254, 281, 303
  T (historical trends) 68
  T (historical trends) 34, 70, 187, 250, 252, 343
using 70, 252
Actions pull-down 22
actions pull-down 148
activating information 50
active workloads 138, 156, 314, 334
adding and releasing sessions 375
adjust the @OII environment 49
  panel appearance 49
time format 49
adjusting @OII environment 47
  panel appearance 49
date format 49
  panel ID 50
adjusting the &OII environment
  panel content
  suspend auto refresh 51
thresholds
  migrating from &O 54
adjusting the &OII environment 53
  panel content
  auto refresh interval 51
/bottleneck analysis 51
default view 51
impact analysis time-out period 51
wait reasons 51
profile maintenance
  copying from another profile 57
creating a new profile 57
deleting a profile 58
printing a profile 58
switching profiles 56
thresholds
  changing settings 54
  defined 53
  disabling 55
  resetting to default value 55
adjusting the @OII environment
  panel appearance 49
adjusting the @OII environment 49, 50
  panel appearance
  current date and time 50
  function key area 50
  realtime password 50
  terminal alarm 50
  time format 49
  trigger key for zoom 50
panel content
  zoom-to-&O profile 51
adjusting thresholds 53, 54
adjusting the &OII environment
  profile maintenance
  specifying different startup profile 56
alert
  key task 301
alert handling 67
alert monitoring 71, 254
alert, key task 124
alerts
  status 250
analysis, bottleneck and impact analysis 138
Analyze Problems action 34, 54, 68, 145, 250
Analyze Problems panel
  I/O wait 85, 266
analyzing DASD problems 107, 286
analyzing existing thresholds 53
and compatibility/goal modes 7
appearance and content of panels 48
APPEND keyword 350
arrow (>) symbol 32
arrows on panel body
double 32
  single 32
ASCII format 222, 350
ASCR
  assigned support center representative 384
 assigned support center representative
  ASCR 384
assigning a function key to a screen space 162
asterisk (*) as a mask 203
asterisk (*) status symbol 67, 250
attribute record 225, 352, 354, 356
auto refresh 34
auto refresh interval 51
Automatic Screen Facility
  .RTN command 176, 178
  /LOG command 176
  commands required 176
  example of 176
  functions in dedicated or VTAM mode 176
  how to use 176
  logging parameters 180
  OPTN command 176
  OUTP major command 180
  report log 178, 180
  screen space example 177
  setting parameters 176
  XACB command 176
  XSUM command 176
automatic sign on 377
automating and logging screen spaces 174
B
B action code 34, 68, 139, 146, 315
backward scrolling 25, 29, 40
BAND keyword 354
bar graphs 196
BATCH
  example of investigating a BATCH problem 272
Batch Job Comparison Report 234, 363
batch job scheduling 195
Batch Job Trending and Forecasting Report 235, 363
Batch Jobs Overview panel
  elapsed time 81, 261
BATCH problem 272
  example, Investigating a BATCH problem 93
Batch Program Resource Consumption Report 235, 363
batch report controls 192
batch reporter for Standard Profile Report 207
BATNAME override 236, 364
beep 50
binary format 222
binary output files for historical data 225, 353
blank status symbol 250
blanks
  as status character 250
blanks, as status character 67
blue (turquoise) status light 252
blue status light 249
bottleneck analysis 138, 314
  enabling 51
bottleneck and impact analysis 314
bottleneck diagnosis 145
Bottlenecks action 34, 68, 146, 315
bottlenecks action 139
brackets (<>) 36
building PROFILE commands 201

C
C (character) data type 224, 226, 352, 355
cache controllers
  historical statistics 110
cache data collection frequency 107, 285
cache memory
  historical statistics 289
  inactive volumes 108, 286
  read/write hit percentages 108, 287
  realtime statistics 109, 288
  setting data-collection frequency 107, 285
  setting thresholds 106, 284
  volumes exceeding thresholds 107, 286
cache statistics 105, 109, 283
cache statistics and data 288
cache volumes
  inactive 286
cache volumes, inactive 108
Candle logo panel 18
causes of workload delay 138
causes of workload delays and bottlenecks 138, 314
chaining screen spaces 169, 170
  .FGO command 171
  .SGO command 171
  .VAR immediate command 171
changing the polling interval 376
Channel I/O problem 95, 115, 274, 293
CHANNEL override 236, 364
Channel Peak Utilization Report 233, 362
Channel Utilization Report 232, 240, 362
color terminal 26
status indicators 249
  color to indicate selectable items 26
COLOR keyword 224, 225, 352, 354
command continuation (-) 202
command mode 159
  comment character (>) 161
comment lines 161
  comment record 225, 354
Common Storage Area (CSA) 101
  active users 281
  Active Users panel 104
  allocated but unowned 281
  common storage utilization 281
  example of approaching limit 280
  growth in use 281
  size of storage 281
  storage area 281
  storage in use 281
  compare workloads to their profiles 209
  comparing workloads to their profiles COMPAR and SETP) 211
Comparison Reports
  Profile/Workload Comparison Report with device
Creating a Comparison Report for multiple system IDs

Creating a Comparison Summary Report showing deviation

Creating a profile display for a specific time interval

Creating a profile display for entire workload type

Creating a profile display for specific system IDs

Creating a profile display with device detail

Creating a profile that excludes atypical records

Creating a screen space

Creating multiple profiles

Creating PROFILE commands

Creating profiles for workloads (PROFILE)

Creating profiles using a mask

Critical condition

Critical status character, controlling

Critical workloads

Identifying

Watching

cross memory collectors synchronization

cross memory mode

cross memory mode segment

cross memory/cross system modes

cross system mode

cross system mode segment

CSA (Common Service Area)

CSA problem

Example, approaching limit

CSAFT command

Current bottleneck

Current date and time

Current problem

Cursor position

Control data collection mnemonics

Control log

OUTP XLFLOG

OUTP command parameters

OUTP major command

OUTP REPORT

When logging occurs

Control log for XLF, ASF, TSF characteristics

Controlling formats

Controlling frequency of collection

Controlling navigation

Controlling the appearance and content of panels

Conventions, panel

Double arrow

Exclamation mark point

Numeric scale

Single arrow

Conventions, panels

Horizontal line

Copying settings from another profile

CPU utilization

Example of investigating a CPU utilization problem

CPU Utilization by Performance Group

CPU utilization problem

Example, Investigating a CPU utilization problem

CPU waits

Troubleshooting

CPU waits, troubleshooting

Creating a comparison report

Detail

Comparison reports

Profile/Workload Comparison Report

Profile/Workload Comparison Summary Report showing deviation

Comparison reports

Profile/Workload Comparison Summary Report showing deviation

Compatibility mode

Compared to goal mode

Panel content

User interface

COMPXT command

Conditional fetching for screen spaces

Conflict

Enqueue

Conflict, enqueue

Conflicts, enqueue and reserve

Content of panels

Contention, resource

Control DASD Display

Control log

Controlling formats

Controlling frequency of collection

Controlling navigation

Controlling the appearance and content of panels

Conventions, panel

Double arrow

Exclamation mark point

Numeric scale

Single arrow

Conventions, panels

Horizontal line

CPU waits

Troubleshooting

CPU waits, troubleshooting

Creating a comparison report

Example of investigating a CPU utilization problem

CPU Utilization by Performance Group

CPU utilization problem

Example, Investigating a CPU utilization problem

CPU waits

Troubleshooting

CPU waits, troubleshooting

Creating a comparison report
customer support
  base maintenance plan 380
  contact information 385
  enhanced support services 384
eSupport 381
  severity levels 381
  telephone support 380

D
D (floating-point) data type 226, 355
DASD exception thresholds 106, 284
DASD problems 286
  107
DASD Response and % Busy 119, 298
DASD Seek Analysis for a Device panel 88, 268
DASD Utilization Report 233
DASD, shared 133, 311
data
  cache 288
data collection frequency 51
data extraction procedure (&USERPROC) 230, 237, 239, 259, 365
DATA keyword 354
data representation
  example 31
  graphic 31
  numeric 31
data to be exported 349
data types 224, 352, 355
dataset
  defining for OBTAIN to use 221
dataset, defining for OBTAIN to use 349
datasets
  link list 289
datasets, link list 110
date and time 183, 337
date format, controlling 49
date/time range 151
DDNAME override 236, 240, 364, 368
deadly embrace 126, 304
DEDICATE statement 377
dedicated mode
  OMEGAMON II 373
default view 51
defaults, session 36, 140
defining a dataset for OBTAIN to use 221, 349
defining meaningful workload profiles 199
defining the data to be exported 221, 349
definition mode 159
degradation, shared-DASD 135
degradation, shared-DASD 311
delay causes
  workload 138
delayed fetching 173
delayed workloads 138, 154, 313, 314, 333
deleting a profile 58
deleting lines 162
deleting lines within 162
Demand Paging Activity Report 233
designing your own historical reports 219
detail date and time 337
details for a Job or Started Task panel 86, 266
developing a resource report 240
developing your own realtime screens
  defined 157
  member of partitioned dataset 157
text editor, using 157
developing your own SAS reports 240
device contention, troubleshooting 154, 333
DEVICE override 236, 240, 364, 368
diagnosing a current bottleneck 145, 320
DIR (director mode segment) 131, 309
director, XMF/XSF
  defined 307
director mode segment 309
display screen 309
in VTAM mode 310
  segment defined 309
disabled condition 250
designing your own historical reports 347
disk fragmentation
  troubleshooting 334
disk fragmentation, troubleshooting 156
DISPLAY command 199, 204, 205
  EPILOG JDAS keyword 133, 311
  profile report for specific time interval 205
  profile report for specific system IDs 206
  profile report for workload type 206
  profile report with device detail 206
  reports (profile report with device detail) 208
  reports (Standard Profile Report) 207
displaying historical detail information 183
displaying profiles (DISPLAY) 205
displaying trends for performance group 150
dollar sign ($) status symbol 67, 250
double arrow (>>) symbol 32
double arrow on panel body 32
downloading historical data 219
  see exporting historical data
DRDY exception 177
DSK (cross system mode segment) 309
DSNAME override 240, 368
dynamic profile update facility 58
E

EBCDIC output files for historical data 223, 352
with delimiters 227, 356
ECB wait 213
ECB wait with STIMER 257
editing batch report controls 192
EDSLIST 349
elapsed time 213
elapsed time problem 261
Elapsed Time Profile area 81, 262
elapsed time problem 80
enabling fast path mnemonics 36, 49
Enqueue 213
enqueue and reserve conflicts 36
enqueue conflict
canceling a job 304
example of responding to
enqueue conflict

cancelling a job 126
guideline for troubleshooting 333
guidelines for troubleshooting 154
responding to 304
enqueue delays 154, 333
enqueues 318
Enterprise Systems Connection 35
environment, adjusting 47
EPILOG PDAS keyword 133
EPPARM 349
EPSYSIN DD statement 351
ESCON 35
eSupport
customer support 381
example of a screen space 160
Exception Logging Facility
logging parameters 180
Exception Logging Facility
.XRN command 175
Exception Logging Facility
.XSUM command 175
/XLF INFOo-line command 175
commands required 175
how to use 175
OPTN command 175
OUTP major command 180
report log 180
setting parameters 175
XACB immediate command 175
XLFLOG log 180
exception thresholds
DASD 284
exception thresholds, DASD 106
exclamation (!) symbol 32
exclamation point on panel body 32
excluding wait reasons from bottleneck analysis 143
execution state 138, 314
exit confirmation 19
exit mnemonics 33
Exit Omegamon II 20
exported data 221, 349
exporting data to SAS 235, 363
exporting historical data 347

OBTAIN command 348
see OBTAIN command
see SET command
extended help 39
EXTRACT command 238, 362, 366
extract data 221, 349

F

F1 - F15 function keys 24
F1 - Help 18
F19 161
F20 161
F4 prompt for selections 29
F7 161
F7 - F8 scrolling keys 29
F8 161
fast path
enabling 36, 49
examples of 35
ge 209
obh 192
using 35
fast path mnemonics 49
fast paths
go 216
iihs 288
iko 304
ils 306
iks 306
iksr 291
iww 315
iwwf 321
ocs 316
ocw 318
ohh 339
ohr 337
fetching screen spaces 169
fetching screen spaces, delayed
.SGO command 173
field level help 38
file names 193
finding inactive cache volumes 108, 286
finding read and write hit percentages 108, 287
finding the problem service class 320
Index
batch reports (accessing controls) 192
batch reports (controls, editing) 192
batch reports (OMIIBATR procedure) 193
controls, online 182
date and time 337
details 338
details, displaying 183
displaying historical details 338
H (historical details) action code 338
history refresh interval 182
online controls 182, 337
online controls (date and time) 183
online controls (detail date and time) 183
online controls (trend date and time) 183
online reports 182, 338, 343
online reports, historical details 183
online reports, historical trends 187
printed reports 190
printed reports (bar graphs) 196
printed reports (detail report example) 196
printed reports (generating) 195
printed reports (interpreting) 196
printed reports (resource reports, types) 191
printed reports (scheduling) 195
printed reports (selecting resource reports) 194
printed reports (selecting workload reports) 193
printed reports (submitting) 195
printed reports (tabular reports) 196
printed reports (workload reports, types) 190
printed reports (detail report example) 196
refresh interval 182, 337
resource reports 191
setting historical controls 337
T (historical trends) action code 343
trend date and time 337
trends 182
trends, displaying 187, 343
trends, TOS response time example 76
workload reports 190
workload reports (interpreting) 196
historical controls
online controls 182
historical data
compatibility mode 181
downloading 219
exporting 219
historical data for cache controllers 110
historical data, exporting
see exporting historical data
historical detail action code 183, 338
Historical Details action 34
Historical Details action 250
historical Details action 68
Historical Details for Batch Job panel 83, 264
historical perspective ??–346
historical reports
customizing 219
historical reports, customizing
see exporting historical data
historical shared-DASD performance 133, 311
historical trend action code 187, 343
historical trend date/time 77
Historical Trends action 34, 68, 76, 150, 250
hit percentages, read and write 108
horizontal line 32
hyphen (-) command continuation 202
hyphen (-) status symbol 67, 250
i (index) mnemonics 33
I (numeric signed integer) data type 226, 355
I command 162
I/O active 213
I/O enqueue 213
I/O problems 312
I/O reserved 213
I/O wait problem 265
I/O wait, troubleshooting 154, 320, 333
icse fast path 36
ID, SMF system 194
identifying bottlenecks and impactors 146
identifying chronic response time problem 150
identifying the greatest impactor 324
identifying the impactor 152
identifying the type of contention 323
idle condition 250
idle waits 318
idle workloads 138, 314
ihs fast path 288
iko fast path 126, 304
iks fast path 306
iks fast path 112, 291
ikss fath 79
illustrating panel format 21
impact analysis 314
impact analysis time-out period 51, 140, 142
impactor identification 152
inactive cache volumes 108, 286
index mnemonics 33
Index pull-down 183, 338, 343
example 186, 189, 342
Index 395

index pull-down 187
Index pull-down 22
Individual Batch Reporting Options pop-up 192
inefficient coding 156
INFO-line commands 161
information on panels
  conventions used 30
  data representation 30
  interpreting 30
information suppression or activation 50
input fields, identifying 26
inserting blank lines 162
interpreting a typical printed report 196
interpreting a typical printed report 196
interpreting Binary Output Files
  FORMAT(INT) 353
interpreting Comparison reports 211
interpreting EBCDIC columnar output files
  FORMAT(COL) 352
interpreting EBCDIC columnar output files with
delimiters
  FORMAT(PC) 356
interpreting information on panels 30
interpreting profile reports 207
interpreting status lights 248
INTERVAL parameter 133, 311
interval, polling 376
investigating a CSA problem 280
investigating a current problem 254
Investigating a DASD problem 295
investigating a key task alert 124, 301
investigating a recurring problem 257
Investigating a STC/APPC problem 96
investigating a STC/APPC problem 276
Investigating a Tape or DASD problem 117
investigating a Tape or DASD problem 296
Investigating the problem
  Investigating a DASD problem 116
  Investigating a STC/APPC problem 96
  investigating a STC/APPC problem 98
invoking a screen space 163
invoking the batch reporter 205
invoking the batch reporter for a comparison report 212
invoking the batch reporter for comparison reports 210
invoking the batch reporter for the Standard Profile
  Report 207
IODEVICE command 208
IODEVICE keyword 206
IPS
  current 271
  IPS, current 90
  isap fast path 79
  ISPF 195
  ispr fast path 79
  iwro fast path 139
  iww fast path 315
  iwwf fast path 321

J
JCL and print parameters 193
JCL submission 195
JCL, saving 195
JDAS keyword 133, 311
job, batch 195
JOBNAME keyword 227, 228, 355, 357

K
k (toolkit) mnemonics 33
KCPCOLOR SAS macro 370
KEPBATCHM macro 242
KEPBATCHM SAS macro 370
KEPCOLLC SAS macro 370
KEPCOLLS SAS macro 370
KEPCOLLS, KEPCOLLC macros 242
KEPCOLOR macro 243
KEPCOLOR override 236, 364
KEPDELGR macro 243
KEPDASD SAS macro 371
KEPEDS dataset 221
KEPMREPL SAS macro 371
KEPPGNMA macro 243
KEPREPL macro 243
KEPSUMMG macro 243
KEPVERSI macro 243
KEPVERSI SAS macro 371
key task
  investigating problem 124, 301
  monitoring critical tasks 124, 301
keys, function 24
keys, scrolling 29
kit, starter (SAS) 361
km2pdmod 373

L
l (control data collection) mnemonics 33
L action code 34, 51, 249
LEAD override 236, 364
Library Lookaside Area (LLA) 292
lights
  associated conditions 249
lights, associated conditions 67
lines that comment themselves out 161
lines, comment 161
lines, separator 161
link datasets
SYS1.PARMLIB members 111
link list datasets 110, 289
and Library Lookaside Area (LLA) 292
and LLA 113
authorization to replace 111, 290
listing all 112, 290
replacing without an IPL 112, 291
SYS1.PARMLIB members 290
LLA 292
LLA (Library Lookaside Area) 113
loading screen spaces from disk to main storage 168
locating the volume 322
logical partition information 35
logon options 18
long-term bottleneck analysis 140, 316
looking at performance group trends 151
looking up cache statistics 105, 283
looping condition
troubleshooting 334
looping condition, troubleshooting 156
looping screen spaces 170
LPAR statistics 35
LSCR immediate command 168
ISystem Status panel 21

M
maintenance assessment service
MAS 384
managing your screen space library 164
manipulating the log within 172
MAS
maintenance assessment service 384
mask (*) 203
MAXJOB override 237, 365
menus
closing 26
eexample 27
using 26
methods for using different profiles 57
mnemonics
action codes 33
declared 33
enabling 36, 49
examples of 33
fast paths 33
i (index) 33
k (toolkit) 33
l (control data collection) 33
using 33
using multiple to create fast path
eamples of 35
using to act upon an object 23
x (exit) 33
mnemonics, fast path 49
mode
compatibility
panel content 17
user interface 17
compatibility (historical data) 181
compatibility, using this document for
mode 8
compatibility, workload and resource
management 7
goal
panel content 17
user interface 17
goal, workload and resource management 7
modes
cross memory/cross system 308
modes, cross memory/cross system 130
Modified Action pull-down for All Channel
Paths 116
Modified Action pull-down for All Channel Paths
Modified Action pull-down for All Channel
Paths 295
modifying a screen space 159
monitoring
alerts 254
CICS region 320
healthy system 70, 252
resources 254
shared-DASD 133, 311
when VTAM is unavailable 373
workloads 254
monitoring a healthy system 70, 252
monitoring historical shared-DASD
performance 133, 311
monitoring performance across systems
historical performance 311
realtime performance 307
monitoring realtime performance across systems 129
monitoring realtime workload against historical
workload profile 218
more + indicator 40
moving about quickly in OMEGAMON II 33
MPL (multiprogramming level) 189
MSM
multi-services manager 384
multiprogramming level (MPL) 189
multi-services manager
MSM 384
multi-tasking workloads 139, 315
effect on resource total percentages 139, 315
MVS 5.2 7
MVS Common Service Area (CSA) 310
MVS mode
finding current 35
MVS service policy
finding current 35
MVS/SP 5
and OMEGAMON II performance data 8
and the user interface 8
MVS/SP 5.2
and historical data 181
M (numeric signed integer) data type 224, 352
n screen spaces 172
navigating in OMEGAMON II 33
new user 25
using action codes 34
using mnemonics 33
navigating in Omegamon II for MVS
using mnemonics 35
using pushbuttons 36
navigation control 50
new features 46
non-color terminal 26
non-VTAM terminals 373
NOPLOT keyword 204
normal condition 249
numeric scale on panel body 32
numeric signed integer data type 224, 226, 352, 355

O
object
acting upon 23
defined 23
object-action 23
alternative to 34
observation record 225, 352, 354, 356
OBTAIN command 219, 220, 347, 349
and goal-mode data 221, 350
defining a dataset for 221, 349
defining output files 221, 349
defining the data to be exported 221, 349
description of 220, 348
eample 221, 224, 227, 228, 349, 353, 356, 357
FORMAT(COL) 222, 223, 350, 352
FORMAT(INT) 222, 225, 353
FORMAT(PC) 222, 227, 350, 356
formats 222, 350
modifying sample files for 220, 349
output files 356
output files for mainframe 223, 225
output files for PC 227
output files, for mainframe 352, 353
sample files for 220, 348
steps in using 220, 348
obtain help
about input fields 38
oco fast path 54
ocp fast path 56
ocpc fast path 57
ocpd fast path 58
ocph fast path 57
ocpp fast path 58
ocps fast path 57
ocs fast path 49, 140, 316
ocw fast path 143, 318
ohb fast path 192
ohh fast path 183, 184, 339
ohr fast path 183, 337
oht fast path 183
okay status character, controlling 49
of fast path 54
OMEGAMON
command mode 159
definition mode 159, 161
monitoring across systems 307
OMEGAMON II
dedicated mode 373
OMIIBATR procedure 193
online controls 182, 337
detail date and time 337
history refresh interval 337
trend date and time 337
online help
extended 38
field level 38
function keys 38
glossary 38
online reports
historical details 338
historical trends 343
op fast path 50
operator alert 71, 254
enqueue conflict example 126, 304
investigating red or yellow light 67, 250
key task example 124, 301
location on System Status panel 248
operator alerts
investigating green light 252
investing green light
statuslights
green or turquoise (blue) 70
operator's toolkit 126, 304
Options pull-down 22
options pull-down 54
OPTN command 176, 179
ot fast path 36
otc fast path 55
otd fast path 55, 197
ote fast path 55
otf fast path 55
other waits 318
oti fast path 55
otm fast path 55, 197
otp fast path 55, 197
otr fast path 55
ots fast path 55
otu fast path 55, 197
otv fast path 55
OUTFILE keyword 221, 349, 353
OUTLIER command 202, 204
output dataset
defining for OBTAIN to use 221
output dataset, defining for OBTAIN to use 349
output filefor OBTAIN 221, 349
output interpretationhistorical printed report contents 196
horizontal scale 197
overcommitted CPU 334

P
P (packed decimal) data type 224, 226, 352, 355
packed decimal data type 224, 226, 352, 355
paging delay 188
paging waits 318
paging waits, troubleshooting 154, 333
panel appearance 49
status characters 49
panel body 21
data representation 30
information on panel 23
interpreting information on 30
panel content 50
compatibility and goal modes 17
in compatibility and goal modes 8
panel formatlayout 21
System Status panel 21
panel help 39
panel ID 50
panels
System Status 248
parameters, DEDICATE 377
parameters, JCL and print 193
parentheses 36
parentheses () 36
PARMCARDS 238, 366
password, supplying realtime 50
PDAS keyword 133
PDS member names (SAS) 362
PDS members 232
percentages
read and write hit 287
percentages, read and write hit 108
PERFINDEX 350
performance across systemsdefined 129
Performance Group Response Time Degradation Report 234, 241
Performance Group Trending and Forecasting Report 234
performance group trends 151
performance groups 145
performance index
description 258
example 258, 320, 344
performance monitoring across systems129, 307–312
historical performance 311
realtime performance 307
period, time-out 142
PGN override 237
PLOT keyword 204
PLOTMIN keyword 227, 228, 352, 355, 357
plus sign (+) 28
plus sign (+) 24
polling interval 376
POP 172
pop-up, session defaults 140
print log 54
print parameters 193
printed historical reports 190, 195
printed reports 196
printing &OII panels 54
setting print options 54
printing a profile 58
printing OMEGAMON II panels 25
problems
CSA 280
current 254
DASD 286
elapsed time 261
recurring 257
service class 320
procedure
graphics replay 238, 366
producing SAS graphical reports 230, 359
productivity index 213
profile
creating 204
PROFILE command 200
profile (WPF)
COMPARE command 209
creating 202
creating (standard profile) 201
creating profile that excludes atypical records 202
defining meaningful 199, 200
DISPLAY command 205
SETP command 209, 211
testing and graphing 204
PROFILE command 199, 201
profiles
creating a new profile 57
deleting a profile 58
dynamic profile update facility 58
printing a profile 58
Profile Maintenance Options menu 56
switching profiles 56
profiles (WPF)
skewed 200
prompt field 24, 28
pull-down, Go To 189
pull-down, Index 186, 189
PUSH 172
pushbuttons
example of 36, 282, 323
location of 36
using 36, 70, 252

Q
QLLA command 111, 290
quiesce request
description 326
using 326

R
RANGE keyword 354
range, date/time 151
read and write hit percentages 108, 287
real storage contention, troubleshooting 154, 333
Real Storage Utilization by Performance Group
Report 233
realtime cache statistics 109, 288
realtime password 50
realtime performance across systems 129
reasons, wait 143, 314, 318
recognizing selectable items 26
record
attribute 225
comment 225
observation 225
recurring bottlenecks 150, 328
recurring problems 140, 316
red status light 68, 69, 249, 250, 251
BATCH example 93, 272
CPU utilization example 122, 299
CSA example 101
key task example 124, 301
TSO response time example 254
refresh data with F5 24
refresh interval 51, 337
historical 182, 337
releasing sessions 375
renaming screen spaces 166
RENM immediate command
arguments 166
replacing link list datasets 110, 289
replay procedure
graphics 366
report
Standard Profile Report 207
REPORT files 179
report log, automatic activation 178
REPORTIF keyword 227, 228, 352, 355, 357
reports (SAS)
graphical 230, 359
historical 347
reserve conflicts 36
resource contention 321
resource detail 191
resource exception 191
resource monitoring 71, 254
Resource status
CPU utilization example 299
resource status 71, 254
CPU utilization example 122
CSA example 101, 280
investigating green light 70, 252
investigating red or yellow light 67, 250
location on System Status panel 248
resource utilization 321
resource waits 154, 333
responding to an enqueue conflict 126, 304
Response
DASD and % Busy 298
response time
distribution graph of poor response time 258
effect of STIMER service 257
example of poor response time 254, 257
poor 257
response time goal
change from average to percentile 261
response time problem, chronic 150
Response, DASD and % Busy
DASD Response and % Busy 119
RIF keyword 354
RKANPAR 221
RK2IN DD statement 222
RK2OUTD 221
RKOMPCSV ddname 159, 165
RKOMPROC ddname 165
RMF interval 182
RMF level 35
RSCLOUT DD statement 353
running OMEGAMON II when VTAM is unavailable 373
running the graphics replay procedure 366

S
S action 72
S action code 34, 68, 69, 124, 250, 251, 252, 254, 257, 281, 303
S km2pbdmod 373
sample of a screen space 160
sample Standard Profile Report 207
SAS 219, 347
SAS graphical reports 230, 359
SAS interface
purpose of 230, 359
requirements 230, 359
SAS macros 242, 370
SAS program parameters 230
data extraction procedure (&USERPROC) 230, 237
user overrides 230
SAS programs 231
data extraction procedure (&USERPROC) 359, 365
displaying results of 231, 360
graphics replay file 360–361
parameters 236, 237, 359, 364, 365
running 231, 238, 360, 366
user overrides 236–237, 364
using macros 242, 370
SAS reports
batch job 234, 363
Batch Job Trending and Forecasting Report, developing 369
batch program 363
Channel Utilization Report 240
data extraction procedure (&USERPROC) 230, 237, 359, 365
developing 240, 369
overview of 364
overview of producing SAS reports 235
performance group 234
Performance Group Response Time Degradation Report 241
producing 230, 236, 237, 359, 364, 365
resource 362
resource type 232
types of 362, 363
user overrides 359, 365
SAS starter kit 235
scheduling the Batch Job 195
screen paces
additional features 169
guidelines 161
screen print 54
output routing options 54
screen spaces 162
screen space example 160
SFC command 170, 173
LOG immediate command 172
.SGO command 170, 173
/REF INFO-line command 159
/SAVE INFO-line command 159
assign F key to screen space chaining together 170
chaining, conditional 171
clear the screen 159
command mode 159
cursor position 161
definition mode 159, 161
fetching, conditional 171
fetching, delayed 173
FGOLOOP keyword 170
INFO-line commands in 161
invoking 163
library, managing 164
library, screen space 164
lines that comment themselves out 161
loading 168
logging features
PUSH argument 172
logging features
POP argument 172
looping 170
LSCR immediate command 168
manipulating the log 172
modifying 158
naming 159
RENM immediate command 166
Index

replacing 159
RKOMPCSV ddname 166
testing .FGO routines 170
SCRN immediate command 165
scrolling 161
backward 25, 29, 40
forward 25, 29, 40
function keys 29
Lines field 30
specifying beginning line number 30
SDATE 354
SEC override 237
see performance index
see = `monitoring’ .performance monitoring 70
selectable items
colors 26
on color terminals 26
on non-color terminals 26
parentheses 26
recognizing 26
underlining 26
selectable items
highlighting 26
SELECTIF keyword 227, 228, 352, 355, 357
selecting date/time range 151
separator lines 161
SEQDATA 350
service class 320
service class resource utilization 321
service objectives 248
service policy 35
session adding and releasing 375
session defaults 36, 140
Session Defaults pop-up 140, 316
SET command 220, 222, 350
description of 222, 350
examples in a command sequence 224, 227, 229,
353, 356, 357
SETP command 199, 211
setting &O logging on/off 60
setting cache data collection frequency 107, 285
setting DASD exception thresholds 106, 284
setting default values for comparisons 211
setting detail date and time 183
setting historical controls 182
setting up and using automatic sign on 377
severity levels
customer support 381
shared DASD performance 311
shared-DASD degradation 133, 311
by workload 311
shared-DASD degradation
by performance group 136
by workload 135
shared-DASD performance 133
Show Details action 34, 68, 69, 72, 124, 147, 250,
251, 252, 254, 281, 303
S action code 70
SIF keyword 354
Sign On panel input fields 18
sign on, automatic 377
signing off Omegamon II
exit confirmation 19
from System Status panel 19
signing onto Omegamon II
Candle logo panel 18
input fields 18
sign on panel 18
single arrow on panel body 32
skewed profiles 200
slash 34
slash (/) 23, 127, 143, 148, 305, 318
SLR 219, 347
SMF system ID 194
solving current problems 140
Solving the STC/APPc problem 99
solving the STC/APPc problem 279
solving workload delays 313
specifying beginning line number 30
specifying file names 193
specifying two or more system IDs 203
specifying two workloads of the same type 202
SRM delay, troubleshooting 154, 333
SRM delays 318
SRM vs. WLM controls 7
start
getting started
user interface 17
start performance monitoring across systems 129
start up CLIST 159
started task PROC 159
starter kit
SAS 235, 361
statistics
cache 283, 288
statistics, cache 105
statistics, realtime cache 109
status alert
for workloads 68
associated conditions 249
for operators 69, 124, 126, 251, 301, 304
for resources 68, 250
for workloads 250
general guide for handling 67, 250
on non-color terminals 249
symbols 249
  associated conditions changing 67, 250
  on non-color terminals
status character, controlling 49
status lights
  associated conditions 249
  green or turquoise (blue) 252
  on color terminals 249
red or yellow 68, 69, 93, 101, 122, 250, 251, 254, 272, 280, 299
red status light
  TSO response time example 72
status symbols 249
STC/APPC problem
  investigating 276
STC/APPC status indicator 184
steps in creating or modifying a screen space 159
STIME 354
STIMER service 257
suppressing or activating information 50
suspend auto refresh 51
sustained peak utilization 362
swap page-in wait 213
Swap Paging Activity Report 233
swaps, troubleshooting 154, 334
symbols
  status 249
  symbols and associated conditions 249
  symbols, status 67
synchronizing cross memory collectors 132
synchronizing XMF collectors 310
SYSIN 349
system configuration information 35
System Environment panel 35
system ID, SMF 194
system performance
  monitoring when VTAM is unavailable 373
system programmer’s toolkit 128, 306
System Resource Manager 7
System Status panel
  Batch status indicator 261, 265
  Channel status indicator 95, 115, 274, 293
  interpreting lights on returning to 25
  using in goal mode 17
System status panel
  Batch status indicator 80
  system status panel 19

T
T (time) data type 226, 355
t action (trends) 23
T action code 34, 68, 187, 250, 343
tabular format 196
t action code 76
tape mounts 318
technical term, defining 40
telephone support
  customer support 380
terminal alarm 50
TERMINAL override 237, 240, 365, 368
terminals
  color 26
  non-color 26
terminals, non-VTAM 373
TEST keyword 204
testing and graphing a profile 204
testing data before it is written to a profile 204
testing data before writing it to a profile 204
testing routines 170
THRESHLD override 237, 365
threshold adjustment 54
threshold analysis 53
thresholds
  accessing Thresholds menu 54
  adjusting 53, 54
  analyzing existing thresholds 53
DASD exception 284
defined 53
  disabling 55
  displaying values of 32
  printing a profile 58
relationship to status lights 248
resetting to default value 55
used in printed resource reports 197
used in printed workload reports 197
thresholds, DASD, 106
thresholds
  migrating from &O 54
time afternoon indicator 49
time and date 183
time data type 226, 355
time format, controlling 49
TIME keyword 354
time morning indicator 49
time separator character 49
time, resetting &OII 61
time/date range 151
Timed Screen Facility (TSF)
  operates with screen auto refresh 179
Timed Screen Facility
  logging parameters 180
  OUTP major command 180
  report log 180
Timed Screen Facility (TSF) 179
.RTN command 179
/LOG INFO-line command 179
commands required 179
how to use 179
OPTN command 179
time-out period 142
time-out period for impact analysis 140
toolkit 126
example of operator’s 126
example of operator’s toolkit 304
system programmer’s 128, 306
toolkit mnemonics 33
TRAP override 237, 365
trend date and time 337
trends 150
Trends for Job or Started Task panel 82, 264
trends, performance group 151
trigger key 50
trigger key for zoom 50
troubleshooting bottleneck guidelines 154
troubleshooting bottlenecks 154
troubleshooting guidelines
active I/O 334
bottlenecks 333
paging waits 333
SRM delay 333
using CPU 334
waiting for CPU 333
TSO response time
investigating a current problem 72
investigating a current problem 76, 257
paging delay 188
poor response time example 72, 254
trend date/time selections 77
using response time distribution 257
turquoise status light 252
tutorial 46
types of data 352
types, wait reason 143

U
U (unsigned integer) data type 355
underlining 26
underlining to indicate selectable items 26
understanding bottleneck and impact analysis 138, 314
understanding the starter kit 361
unsigned integer data type 226, 355
user information 46
user interface
customizing 47
goal mode
user interface 17
goal mode vs. compatibility mode 8, 17
understanding 21
user overrides 230, 236–237, 359, 364, 365
using action codes 34
using automatic sign on 377
using CPU 213
using different profiles 56, 57
using DISPLAY to generate online reports 205
using mnemonics 33
using multiple mnemonics to create a fast path 35

V
view problems (vp) 184
View pull-down 22
view, defaultl 51
viewing historical data for cache controllers 110
viewing realtime cache statistics 109, 288
volume location 322
volumes
inactive cache 286
volumes, inactive cache 108
vp (view problems) 184
VSAM dataset 198
VTAM telecommunications failure
running OMEGAMON II during 373

W
Wait Reason 213
Wait Reason Reporting Control pop-up 143, 318
wait reason types 143
wait reasons 314
active I/O 156, 334
definition 138, 314
ECB wait with STIMER 257
tenqueue delays 154, 333
excluding from bottleneck analysis 143, 318
paging waits 154, 333
selecting 51
SRM delay 154, 333
using CPU 156, 334
Wait Reason Report Control pop-up 143
Wait Reason Reporting Control pop-up 318
waiting for CPU 154, 333
waiting for I/O 154, 333
waiting for CPU 154, 213, 334
waiting for I/O 154, 333
waiting for MVS lock 213
waiting for resources 154, 333
warning condition 249
warning status character, controlling 49
watching critical workloads over time 198
what causes a workload delay 138
windows
  closing 29
  example 27
  learning about 27
  uses 27
WLM mode, current 35
WLM vs. IPS controls 7
workload delay, causes 138
workload delays and bottlenecks 313–334
  active I/O 334
  active workload defined 138, 314
  affecting a CICS region 320
  analyzing a recurring bottleneck 150, 328
  bottlenecks defined 138, 314
  Bottlenecks panel, navigation to 315
  bottlenecks panel, navigation to 139
  collecting bottleneck and impact analysis information 140, 316
  delayed workload defined 138, 314
  diagnosing a current bottleneck 145, 320
  enqueue delays 333
  excluding wait reasons 143, 318
  execution states 138, 314
  historical scenario 150
  identifying causes 79
  idle workload defined 138
  idle workloads defined 314
  impact analysis defined 138, 315
  paging waits 333
  realtime scenario 145, 320
  recurring problems 140, 316, 328
  resource group capping 328
  Session Defaults pop-up 316
SRM delay 333
troubleshooting guidelines 154, 333
  enqueue delays 154
  paging waits 154
  save I/O 156
  SRM delay 154
  using CPU 156
  waiting for CPU 154
  waiting for I/O 154
understanding bottleneck and impact analysis 138, 314
  using CPU 334
  using historical displays 328
  wait reasons defined 138, 314
  waiting for CPU 333
  waiting for I/O 333
workload detail 191
workload exception 191
Workload Manager 321
  compared to ISP controls 7
workload monitoring 71, 254
Workload Profiling Facility (WPF)
  comparing workloads to their profiles 209
  comparing workloads too their profiles 211
Workload Profiling Facility 198
Workload Profiling Facility (WPF)
  creating profiles for workloads 200
  creating profiles for workloads (PROFILE) 204
  displaying profiles (DISPLAY) 205
  purpose of 199
workload reports 190
Workload status
  BATCH example 93, 272
workload status 71, 254
  investigating green light 252
  investigating red or yellow light 67, 250
  investing green light 70
  location on System Status panel 248
  TSO response time example 72, 254
workload summary 190
workload, critical
  identifying 199
  watching 198
workloads, multi-tasking
  effect on resource total percentages 139, 315
workloads, multitasking 139, 315
WPF (Workload Profiling Facility) 198
WPF (Workload Profiling Facility)
  purpose of 199
WPF command 216
write and read hit percentages 108, 287

X

x (exit) mnemonics 33
X (hexacecimal) data type 355
X (hexadecimal) data type 226
x....y symbol 32
XACB command 218
XACB immediate command 175, 176, 177
XLFLOG files 179
XMF
  see Cross Memory Facility
XMF/XSF
  defined 129
XMM (cross memory mode segment) 309
XSF
  see Cross System Facility
XSUM command 176
Y
yellow status light  68, 69, 249, 250, 251
 CSA example  280

Z
zoom key  50
zoom-to-&O profile  51