# Contents

**List of Figures** .............................................................. 5

**Preface** ............................................................................. 7
- About this book ................................................................. 8
- Adobe Portable Document Format ........................................ 10
- Documentation Conventions .............................................. 11

**What's New** ...................................................................... 13
- Application Trace Facility .................................................... 14
- New OMEGAMON II Functions ............................................ 15
- Documentation Changes ..................................................... 16

**Chapter 1. Bottleneck Analysis Menu Screens** ......................... 17
- Accessing the Bottleneck Analysis Menu Screens .................... 18
- Accessing the Bottleneck Analysis Menu ................................. 19
- Accessing the Factors Affecting Executing Threads Screen ........ 20

**Chapter 2. Bottleneck Analysis Collector** ................................. 23
- About the Bottleneck Analysis Collector ................................. 24
- Attaching the Collector ....................................................... 26
- Controlling the Collector ..................................................... 29
- Specifying Collector Options ............................................... 33

**Chapter 3. Bottleneck Analysis Displays (MDEX/PDEX)** ............ 43
- Displaying Bottleneck Analysis Data ...................................... 44
- Examples of MDEX and PDEX Output ................................. 46
- Factors That Influence the Displays ...................................... 48

**Chapter 4. Bottleneck Analysis Execution States** ...................... 51
- About Execution States ....................................................... 52
- CPU Usage ........................................................................ 53
- Database I/O Waits ............................................................ 55
- MVS Waits ....................................................................... 56
- IMS Waits ....................................................................... 61

**Appendix A. Command Summary** ........................................ 69
- Bottleneck Analysis Commands ........................................... 70
- PSB Group Commands ......................................................... 72
List of Figures

FIGURE 1. Bottleneck Analysis Menu ..................................................... 19
FIGURE 2. Executing Threads Panel ...................................................... 20
FIGURE 3. Attaching the DEXAN Collector Using IDEG .......................... 26
FIGURE 4. IDEG and BEGN Commands Output ...................................... 29
FIGURE 5. Using BEGN from the Menu Interface .................................. 30
FIGURE 6. IDEG and END Commands Output ...................................... 30
FIGURE 7. Using END from the Menu Interface ..................................... 31
FIGURE 8. DTCH Command Output ..................................................... 31
FIGURE 9. Using DTCH from the Menu Interface .................................. 32
FIGURE 10. BMPXOFF and BMPXON Commands Output ...................... 33
FIGURE 11. Using BMPX from the Menu Interface ................................ 34
FIGURE 12. CLRLnn and CLRL Commands Output ............................... 35
FIGURE 13. Using CLRL from the Menu Interface ................................. 35
FIGURE 14. CLRSnn and CLRS Commands Output ............................... 36
FIGURE 15. Using CLRS from the Menu Interface ................................. 36
FIGURE 16. Changing the Database Sampling Option (DBSW) .............. 37
FIGURE 17. Using DBSW from the Menu Interface ............................... 38
FIGURE 18. SUSP Command Output .................................................. 38
FIGURE 19. RESM Command Output .................................................. 39
FIGURE 20. STIM Command Output ................................................... 39
FIGURE 21. Using STIM from the Menu Interface ................................ 40
FIGURE 22. MTHR Command Output ................................................ 40
FIGURE 23. MTHR and SCAL Commands Output ................................ 41
FIGURE 24. THRS Command Output ................................................ 41
FIGURE 25. Using THRS from the Menu Interface ................................ 42
FIGURE 26. ABCD Command Output ................................................. 42
FIGURE 27. MDEX/PDEX Commands Format ..................................... 44
FIGURE 28. MDEX Command Output ................................................ 46
FIGURE 29. PDEX Command Output ................................................ 47
Introduction

This manual contains reference information about the Bottleneck Analysis component of OMEGAMON II® for DBCTL.

In this manual you will find information on the following topics:

- how to invoke Bottleneck Analysis through the OMEGAMON II menu interface and command interface.
- how to control the Bottleneck Analysis collector, the component that samples the degradation in your IMS system.
- how to display bottleneck analysis information in the menu and command interfaces.
- an explanation of how Bottleneck Analysis groups transactions that are waiting for processing, and a description of each reason bottleneck analysis displays when a transaction is waiting to be processed.

This manual does not include information about bottleneck analysis data displayed in the CUA™ interface. For information about all the data displayed in the CUA interface, see the OMEGAMON II for DBCTL User’s Guide.

For information about time- and event-driven features (automatic screen facility, timed screen facility, and exception logging facility), see the OMEGAMON II for IMS Realtime Commands Reference Manual. For information about bottleneck analysis parameters and startup options, refer to Installing Candle Products on MVS.
About this book

Who should use this guide

This manual is for users who are familiar with performance monitoring software. It assumes that you are familiar with OMEGAMON II, and know how to use its menu and command interfaces.

The Bottleneck Analysis component of OMEGAMON II makes the diagnostic power of bottleneck analysis available to those responsible for the performance of complex IMS environments. Bottleneck Analysis is a Candle-developed analysis technique that focuses on workloads rather than resources. It breaks the execution time of workloads into its component parts, allowing performance analysts and systems programmers to focus on the areas most significant to IMS system performance.

Documentation set information

The documentation listed in the following table is available for the Candle IMS Products. To order additional product manuals, contact your Candle Support Services representative. Where to look for more information

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC51-6057</td>
<td>Installing Candle Product on MVS</td>
<td>Provides installation instructions and other installation considerations.</td>
</tr>
<tr>
<td>ID53-6341</td>
<td>OMEGAMON II for DBCTL Realtime Commands Reference Manual</td>
<td>Describes in detail all of the features of the OMEGAMON II for DBCTL command interface.</td>
</tr>
<tr>
<td>ID53-6344</td>
<td>OMEGAMON II for DBCTL Bottleneck Analysis (DEXAN) Reference Manual</td>
<td>Provides reference information and descriptions of the features of the bottleneck analysis component.</td>
</tr>
<tr>
<td>ID53-6345</td>
<td>OMEGAMON II for DBCTL Historical Component (EPILOG) Reference Manual</td>
<td>Provides a comprehensive description of the features of the historical component (EPILOG).</td>
</tr>
<tr>
<td>ID53-6346</td>
<td>OMEGAMON II for DBCTL Historical Component (EPILOG) User’s Guide</td>
<td>Teaches you, step-by-step, how to operate the historical component (EPILOG) reporter after installation.</td>
</tr>
</tbody>
</table>
About this book

For more information related to your product, please see the
- technical documentation CD-ROM that came with your product
- technical documentation information available on the Candle Web site at www.candle.com
- online help provided with your product

Ordering additional documentation

To order additional product manuals, contact your Candle Customer Support representative.

We would like to hear from you

Candle welcomes your comments and suggestions for changes or additions to the documentation sets. 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 DBCTL Bottleneck Analysis Reference Manual Version 510” in the subject line.
Printing this book

Candle supplies documentation in the Adobe Portable Document Format (PDF). The Adobe Acrobat Reader will print PDF documents with the fonts, formatting, and graphics in the original document. To print a Candle document, do the following:

1. Specify the print options for your system. From the Acrobat Reader Menu bar, select **File > Page Setup...** and make your selections. A setting of 300 dpi is highly recommended as is duplex printing if your printer supports this option.

2. To start printing, select **File > Print...** on the Acrobat Reader Menu bar.

3. On the Print pop-up, select one of the **Print Range** options for
   - All
   - Current page
   - Pages from: [ ] to: [ ]

4. (Optional). Select the Shrink to Fit option if you need to fit oversize pages to the paper size currently loaded on your printer.

Printing problems?

The print quality of your output is ultimately determined by your printer. Sometimes printing problems can occur. If you experience printing problems, potential areas to check are:

- settings for your printer and printer driver. (The dpi settings for both your driver and printer should be the same. A setting of 300 dpi is recommended.)
- the printer driver you are using. (You may need a different printer driver or the Universal Printer driver from Adobe. This free printer driver is available at www.adobe.com.)
- the halftone/graphics color adjustment for printing color on black and white printers (check the printer properties under **Start > Settings > Printer**). For more information, see the online help for the Acrobat Reader.
- the amount of available memory in your printer. (Insufficient memory can cause a document or graphics to fail to print.)

For additional information on printing problems, refer to the documentation for your printer or contact your printer manufacturer.

Contacting Adobe

If additional information is needed about Adobe Acrobat Reader or printing problems, see the Readme.pdf file that ships with Adobe Acrobat Reader or contact Adobe at www.adobe.com.
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.

Note: In ordinary text, variable names appear in italics.
Symbols

The following symbols may appear in command syntax:

**Table 2. Symbols in Command Syntax**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td>[ ]</td>
<td>Denotes optional arguments. Those arguments not enclosed in square brackets are required. Example: <strong>APPLDEST DEST [ALTDEST]</strong> In this example, DEST is a required argument and ALTDEST is optional.</td>
</tr>
<tr>
<td>{ }</td>
<td>Some documents use braces to denote required arguments, or to group arguments for clarity. Example: **COMPARE {workload} - REPORT={SUMMARY</td>
</tr>
<tr>
<td>_</td>
<td>Default values are underscored. Example: **COPY infile outfile - [COMPRESS={YES</td>
</tr>
</tbody>
</table>
Chapter overview

Version 510 of OMEGAMON II for IMS and OMEGAMON II for DBCTL significantly enhanced the Application Trace Facility. This version also provides several new functions, which broaden the overall functionality of OMEGAMON II for IMS and OMEGAMON II for DBCTL.

Chapter contents

Application Trace Facility .................................................. 14
New OMEGAMON II Functions ......................................... 15
Documentation Changes ................................................... 16
Application Trace Facility

Application Trace Facility (ATF) is a monitoring agent in OMEGAMON II for IMS and OMEGAMON II for DBCTL. In Version 510, ATF was significantly enhanced so that:

- Multiple ATF OMEGAMON Classic address space sessions can monitor the same IMS
- The IMS Monitor can run concurrently with these ATF sessions
- All environments for IMS, IMS DB/DC, IMS DC and IMS DBCTL are supported
- A site has external control of its operations
- IMS Version 7 DC Monitor is supported
- Concurrent Online TRF display and ATF display functions are supported

In the previous Version 500, ATF had a DETAIL parameter that could be set to LOW or HIGH. In Version 510, this parameter was removed and the function was separated to display this information on separate sets of panels:

- What used to be DETAIL=LOW in ATF V500 is now the Online TRF Display
- What used to be DETAIL=HIGH in ATF V500 is now new ATF panels

The changes made to ATF in this release are explained in detail in the Application Trace Facility Manual for OMEGAMON II for IMS and DBCTL 510. ATF's online help has been upgraded to reflect these new features.
New OMEGAMON II Functions

Several new functions were added to OMEGAMON II for IMS and OMEGAMON II for DBCTL. These functions are:

- Expanded generic IMS command support
- Enhanced VSAM buffer pool statistics
- Enhanced fast path buffer pool statistics
- Enhanced fast path statistics
- Enhanced operator assistance for fast path areas
- Additional data and sorting on IMS Message region fields
Online documentation

With version 510, Candle Corporation has moved OMEGAMON II for IMS manuals from IBM 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 IMS 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 being 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 describes the Bottleneck Analysis menu screens.

Chapter contents

Accessing the Bottleneck Analysis Menu Screens ................. 18
Accessing the Bottleneck Analysis Menu ......................... 19
Accessing the Factors Affecting Executing Threads Screen ........ 20
Accessing the Bottleneck Analysis Menu Screens

Introduction

The Bottleneck Analysis menu screens available through the OMEGAMON II Menu interface provide an easy-to-use format for working with the Bottleneck Analysis component of OMEGAMON II. The chapters that follow explain how Bottleneck Analysis works and how you can customize the displays to obtain data more specific to your IMS environment to help you monitor and tune your system.

To start Bottleneck Analysis automatically when you start OMEGAMON II, follow the CICAT installation procedures in the OMEGAMON II Configuration and Customization Guide. In the section on “Preparing the Startup Files”, specify “Autostart RTM Components” on the “Specify RTM Configuration Values” panel.
Accessing the Bottleneck Analysis Menu

Introduction

This section provides detail steps on how to access the Bottleneck Analysis menu.

Procedure

You can access Bottleneck Analysis from the OMEGAMON II Menu interface by following these steps:

1. From the OMEGAMON II CUA interface System Overview panel, select CMD/Menu Interface (OMEGAMON®) from the GoTo pull-down. OMEGAMON II displays the Zoom to OMEGAMON pop-up so that you can zoom to the Menu Interface screen space.

2. Press Enter to zoom to the default screen space. OMEGAMON II displays the Main Menu.

3. Select BOTTLENECKS from the Main Menu and press Enter. The Bottleneck Analysis menu displays as shown in the following figure.

FIGURE 1. Bottleneck Analysis Menu
Accessing the Factors Affecting Executing Threads Screen

Introduction

This section provides detail steps on how to access the Factors Affecting Executing Threads screen.

Procedure

From the Bottleneck Analysis menu, select option A, EXECUTING and press Enter.

A screen similar to the one in the following figure displays. This screen displays a graph that represents the percentage of time an average thread spent in each of the execution states that the Bottleneck Analysis data collector monitors.

**FIGURE 2. Executing Threads Panel**

Note: See “Bottleneck Analysis Displays (MDEX/PDEX)” on page 43 for more information on Bottleneck Analysis displays.
If you are using Bottleneck Analysis through the OMEGAMON II Menu system, the collector is attached and sampling begins at startup. If your installation has altered the startup parameters for Bottleneck Analysis, and the Bottleneck Analysis collector has not been attached, type a hyphen (-) in column 1 to the left of IDEG to attach the collector. See “Attaching the Collector” on page 26 for an explanation of how to attach the Bottleneck Analysis collector and begin collecting data.
Chapter overview

This chapter provides information about the Bottleneck Analysis collector.

Chapter contents

About the Bottleneck Analysis Collector ......................... 24
Attaching the Collector ........................................ 26
Controlling the Collector ....................................... 29
Specifying Collector Options .................................. 33
About the Bottleneck Analysis Collector

Introduction

The OMEGAMON II interface attaches the Bottleneck Analysis collector as an MVS subtask. At a user-defined interval, the collector samples the processing states of all threads and BMPs running in IMS and records the data it gathers in virtual storage. A processing state (also called an execution state) is either what a thread is doing or why it is waiting. "Using CPU in IMS," "Waiting for CPU," and "IRLM Conflict Wait" are typical execution states. "Bottleneck Analysis Execution States" on page 51 describes execution states in more detail.

The collector runs as a subtask of the interface. You only need one collector (and Bottleneck Analysis permits only one), no matter how many users log on to Bottleneck Analysis. This means that all users access the same data.

Collector Operation

Once the collector has been attached by the OMEGAMON II interface and begins to sample, it allocates virtual storage to hold the data it gathers. This area, often referred to as the counters or buckets, is divided into two parts: the short-term area and the long-term area. The default parameters include an option to allocate space to hold database I/O wait information by individual database name. When this option is invoked, the collector allocates an additional area, which it also divides into short-term and long-term sections.

The data areas contain counters for overall thread activity, and also for activity broken down by PSB group. Bottleneck Analysis supports up to 30 PSB groups. You specify to which groups a given PSB belongs. You can use the PSB groups to target analysis to specific workloads, such as threads which used a certain PSB.

Once the collector allocates the data areas, it begins observing threads running in your IMS system and recording their execution states in its counters. The default is to sample every half-second. You can reduce this setting to as little as every tenth of a second. Very few users run the collector more slowly than one sample per second. The purpose of sampling rapidly is to gather a statistically significant number of samples in a short period of time.

At user-specified intervals, the collector “throws away” the data it collected and starts fresh by zeroing its counters. It does this to prevent the data from being averaged out over such a long period of time that it ceases to be of interest to the user of a realtime monitor. The collector supports two clear intervals, corresponding to its short-term and long-term data areas:

- The short-term interval is intended to be set so the data you see from the short-term counters is from the very recent past. Typically, the short-term interval is five minutes or less. Candle ships Bottleneck Analysis with a default short-term interval of 5 minutes.
The long-term interval is intended to be set so the data you see from the long-term counters gives you a longer perspective on the performance of the system you are studying. Candle ships Bottleneck Analysis with a default long-term interval of 30 minutes.

When the long-term interval expires, Bottleneck Analysis clears both the long-term and short-term buckets. The collector continues to gather data until it is told to stop. If more than one display controller is running under the OMEGAMON II interface, the collector responds to commands from them in the order they are issued. The collector stops gathering data and frees its bucket areas when it receives an END command.
Attaching the Collector

Introduction

By default, the Bottleneck Analysis collector is attached and data collection begins when you start up the OMEGAMON II interface. If your site has modified the startup parameters for Bottleneck Analysis and the collector is not attached, you can attach the Bottleneck Analysis collector in one of these ways:

- from OMEGAMON II with the -IDEG command
- when OMEGAMON II initializes with the START DEXAN command
- from the MVS console

For more information on beginning data collection when you attach the collector, see “Starting data collection when OMEGAMON II initializes” on page 27.

Note: You can also start the collector from the OMEGAMON II System Overview panel in the CUA interface. For more information, see the OMEGAMON II for DBCTL User’s Guide.

Attaching the Collector from OMEGAMON II

To use the -IDEG command to attach the collector from OMEGAMON II:

1. Select option B, CONTROL, from the Bottleneck Analysis menu.
   OMEGAMON II displays a panel where you start and stop the Bottleneck Analysis collector.

FIGURE 3. Attaching the DEXAN Collector Using IDEG

---

---

---

---

---

---

---

---

---

---

---
2. Attach the collector by placing a hyphen (-) in column 1 next to the IDEG command and press Enter, as shown in the previous figure. Instead of using the menus to start the collector, you can type the -IDEG command in command mode.

The -IDEG command attaches the Bottleneck Analysis collector using an internally generated START DEXAN OMEGAMON II interface command. This command does not begin sampling, however. To begin sampling, see “Beginning data collection (BEGN)” on page 29.

Attaching the Collector when OMEGAMON II Initializes

To attach the Bottleneck Analysis collector when OMEGAMON II initializes, issue the following OMEGAMON II interface command in the RKANPAR member, KOIDEXmp (by default this member is executed by KOImpP00).

```
START DEXAN GLOBAL=mp
```

The GLOBAL= parameter specifies which RKANPAR(KOIGBL) load module MVS is to load to supply Bottleneck Analysis parameter defaults. This load module also supplies the default PSB groups. If you do not specify the GLOBAL= parameter, it defaults to M0 and MVS loads and uses the Candle-supplied TKANPAR(KOIGBLmp) load module during initialization. It is recommended that you create your own RKANPAR(KOIGBLmp) load module according to the directions in the Configuration and Customization Guide, in order to collect information on your site’s workload groups.

Starting data collection when OMEGAMON II initializes

If you add the EXEC KOIDEXmp OMEGAMON II interface command to the RKANPAR(KOImpP00) member, you can use the IDEG=BEGN keyword to set the collector to begin sampling immediately after the Bottleneck Analysis collector is attached.

By default, the collector attaches and begins to collect data at OMEGAMON II interface startup when the IDEG=BEGN keyword is included with the START DEXAN command in the RKANPAR(KOIDEXmp) member. If your site has commented out the EXEC KOIDEXmp command, you can issue the START DEXAN or EXEC KOIDEXmp command to the OMEGAMON II interface from the MVS operator’s console.

To begin data collection immediately after attaching the collector, issue the following command to the OMEGAMON II interface:

```
MODIFY <MPPREFIX><IMSID>,START DEXAN,IDEG=BEGN,GLOBAL=mp
```
or modify the EXEC KOIDEXmp statement.
The GLOBAL= parameter specifies which RKANPAR(KOIGBLmp) OMEGAMON II load module MVS should load to supply Bottleneck Analysis parameter defaults. This load module also supplies the default PSB groups. If you do not specify the GLOBAL= parameter, it defaults to M0 and MVS loads and uses TKANPAR(KOIGBLmp) during initialization. It is recommended that you follow the customization process described in the Configuration and Customization Guide, which instructs you not to load the TKANPAR(KOIGBLmp) module.

**Attaching the collector at an MVS console**

You can also enter the START DEXAN command at an MVS console, as in the following example:

   ```sql
   MODIFY <MPREFIX><IMSID>,START DEXAN GLOBAL=mp
   ```

MPREFIX is the two-character modify prefix specified in your startup PROC during installation. The default is M0. IMSID is the four-character subsystem name defined in the IMS SYSGEN. See the Configuration and Customization Guide for information. Again, you can use the GLOBAL= parameter to specify the RKANPAR(KOIGBLmp) OMEGAMON II load module for MVS to load.
Controlling the Collector

Introduction

When you attach the collector, it initializes itself so it can receive control commands. The collector does not begin sampling at this time; it waits for further instructions from Bottleneck Analysis.

You can issue instructions to the collector by typing commands in command mode or by using the Menu interface panels.

IDEG command

Before you can tell the collector what to do in command mode, you must first issue the IDEG major command. You can type the IDEG major command in command mode as follows:

    IDEG

Then, you can issue minor commands of IDEG to control the collector as described in the following sections.

Note: Any Bottleneck Analysis minor command must be preceded by the IDEG major command.

Alternatively, you can use the Menu interface to issue IDEG minor commands. When you use the Menu interface, the IDEG major command automatically precedes any minor command you select.

The following sections describe the IDEG minor commands that you can use to control the collector.

Beginning data collection (BEGN)

The BEGN minor command tells the collector to finish initialization and begin sampling. You can type the BEGN command in command mode as shown in the following figure.

**FIGURE 4. IDEG and BEGN Commands Output**

```
IDEG   >> Elapsed time= 33 SEC #samples(short)=7, #samples(long)=71 <<
begn   >> Collector now active <<
```

This figure shows that the collector started 33 seconds ago.

In the Menu Interface:

1. Start the collector by selecting option **B, CONTROL**, from the Bottleneck Analysis menu.
Controlling the Collector

2. Remove the > character from before the BEGN command and press Enter to begin data collection.
   The panel is updated as shown in the following figure.

FIGURE 5. Using BEGN from the Menu Interface

```plaintext
> Help PF1
> Back PF3
> Up PF7
> Down PF8

Start/Stop the DEXAN Collector and Control Data Collection

> To start the collector, enter a hyphen (-) preceding IDEG.
-IDEG  >> Elapsed time= 16 SEC, #samples(short)=34, #samples(long)=34 <<

> To begin data collection, remove the > preceding BEGN.
>begn  >> Collector now active <<

> To suspend data collection, remove the > preceding END.
>end

> To stop the collector, remove the > preceding DTCH.
>dtch
```

Some of the collector control commands this manual documents can only be issued when the collector is not sampling (either the collector has not received a BEGN command or it has received an END command to halt sampling). An active collector cannot accept these commands because they affect the way the collector allocates the data areas it creates at BEGN time. You can issue any control commands which Bottleneck Analysis must receive before beginning collection between IDEG and BEGN in command mode or before issuing the BEGN command from the Menu interface.

Ending data collection (END)

The END minor command signals the Bottleneck Analysis collector to stop sampling. The collector remains attached, but dormant, waiting for a BEGN command to tell it to start collecting information again. The following figure illustrates using the END command in command mode.

FIGURE 6. IDEG and END Commands Output

```plaintext
IDEG  >> Elapsed time= 6:37 MN, #samples(short)=185, #samples(long)=7 36 <<
>end  >> Collector now dormant <<
```

Use the END command to stop the collector when you want to change some of the options, such as the database I/O reporting option. ("Database sampling (DBSW)" on page 37 describes how this is done with the DBSW command.)
In the Menu Interface:

1. End data collection by selecting option B, **CONTROL**, from the Bottleneck Analysis menu.

2. Remove the > character from before the END command and press Enter to stop data collection.

The panel updates as shown in the following figure.

**FIGURE 7. Using END from the Menu Interface**

---

Detaching the collector from **OMEGAMON II (DTCH)**

The DTCH minor command detaches the Bottleneck Analysis collector from the interface. If the collector is active at the time you enter the DTCH command, Bottleneck Analysis stops it and then detaches it. This command has the same effect as the interface modify command STOP ID=DX, entered at an MVS console.

To detach the collector, you can issue the DTCH minor command of the IDEG major command, as illustrated in the following figure.

**FIGURE 8. DTCH Command Output.**

---

In the Menu Interface:

1. Detach the collector by selecting option B, **CONTROL**, from the Bottleneck Analysis menu.

2. Remove the > character from before the DTCH command and press Enter to detach the collector.

The panel updates as shown in the following figure.
Controlling the Collector

FIGURE 9. Using DTCH from the Menu Interface

After it executes, the DTCH command comments itself out by setting its label field to >.

To attach the collector again, enter the IDEG command with a hyphen in column 1, as described in “Attaching the Collector” on page 26.

Other methods for detaching the collector

You can also detach the collector using one of the following methods:

- Issue the STOP ID=DX interface command to the OMEGAMON II interface.
- Issue the STOP ID=DX OMEGAMON II interface command at the MVS console, as in the following example:

  MODIFY <MPREFIX><IMSID>,STOP ID=DX

  MPREFIX is the two-character modify prefix specified in your startup PROC during installation. The default is M0. IMSID is the four-character subsystem name defined in the IMS SYSGEN. See the Configuration and Customization Guide for information.

- Shut down IMS (this terminates the entire OMEGAMON II address space).
Specifying Collector Options

Introduction

This section describes how to specify options for the collector using IDEG minor commands or the Menu interface. It describes how to perform the following tasks:

- exclude PSBs based on whether they are being processed in batch message processing regions
- set the long-term clear interval
- set the short-term clear interval
- control whether the collector collects data by individual database name
- suspend the collector
- resume collector sampling
- set the sampling interval
- display the threshold for the MDEX command
- display the scaling factor for the MDEX command
- display the threshold for the PDEX command
- display information on collector abends

BMP Thread Sampling (BMPX)

You may want to exclude PSBs that are being processed in batch message processing regions (BMPs) from analysis, since a terminal user is not waiting for a response from this type of PSB. The BMPXON command tells the collector to ignore threads being processed in BMPs. BMPXOFF tells the collector to gather data about threads being processed in non-message-driven BMP regions. Either command forces the collector to clear both the long-term and short-term data gathered so far, and then places the comment character (>) in its label field so Bottleneck Analysis does not re-execute it.

The following label field shows output from BMPXOFF and BMPXON commands.

FIGURE 10. BMPXOFF and BMPXON Commands Output

Specify BMPX (without ON or OFF) to display the current setting. BMPX followed by a question mark (BMPX?) also displays the current setting. If you do not set this option using the BMPX command, Bottleneck Analysis uses the default in the Rkanpar(KOIBLmp) module you use (BMPXOFF in the TKANPAR(KOIBLmp) module that Candle supplies.)
Specifying Collector Options

In the Menu Interface:

1. Specify BMP thread sampling by selecting option **C, OPTIONS**, from the Bottleneck Analysis menu.
2. Remove the > character from before the BMPX command.
3. Add **OFF** or **ON** to the BMPX command and press Enter.

   The panel is updated as shown in the following figure.

**FIGURE 11. Using BMPX from the Menu Interface**

You cannot change the BMPX option while the EPILOG historical component of OMEGAMON II is active. See the OMEGAMON II for IMS Historical Component (EPILOG) Reference Manual for information on stopping EPILOG.

Long-term clear interval (CLRL)

The CLRL command sets the interval for clearing the long-term data counters to \( nn \) minutes. Bottleneck Analysis accepts a value within the range of 1 and 480 minutes (8 hours). A reasonable value for the long-term clear interval is 30 minutes (CLRL30). The value you enter for CLRL must be longer than the short-term clear interval set by CLRS. When you change the value of CLRL, the collector clears both the short-term and long-term counters, and CLRL places the comment character (>) in its label field so Bottleneck Analysis does not re-execute it.

Enter CLRL (without a numerical argument) to clear both the short-term and long-term counters without changing the clear interval.

Figure 12 on page 35 shows two CLRL commands, one with an argument and the other without.
**FIGURE 12. CLRLnn and CLRL Commands Output**

<table>
<thead>
<tr>
<th>Command</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLRL30</td>
<td>LONG-INTERVAL COUNTERS WILL BE RESET EVERY 30 MINUTES</td>
</tr>
<tr>
<td>CLRL</td>
<td>BOTH SHORT AND LONG COUNTERS HAVE BEEN RESET</td>
</tr>
</tbody>
</table>

The CLRL? command displays the existing long-term clear interval. If you do not set this option via the CLRL command, Bottleneck Analysis uses the default in the RKANPAR(KOIGBL) OMEGAMON II load module you use. (This is CLRL30 in the TKANPAR(KOIGBLmp) module Candle supplies.)

In the Menu Interface:

1. Set the interval for clearing the long-term data counters by selecting option **C, OPTIONS**, from the Bottleneck Analysis menu.
2. Remove the > character from before the CLRL command.
3. Add a numeric argument and press Enter.

The panel is updated as shown in the following figure.

**FIGURE 13. Using CLRL from the Menu Interface**

---

**Short-term clear interval (CLRS)**

The CLRS command sets the interval for clearing the short-term data counters to \( nn \) minutes. Bottleneck Analysis accepts a value within the range 01 to 99. The value you enter for CLRS must be shorter than the long-term clear interval set with CLRL. A reasonable value for the short-term clear interval is five minutes (CLRS05). Use at least 30 samples for statistical significance; how long it takes to get them depends on the setting of the cycle time.
When you change the value of CLRS, Bottleneck Analysis clears both the short-term and long-term counters, and CLRS places the comment character (>) in its label field so Bottleneck Analysis does not re-execute it.

Specify CLRS (without a numerical argument) to clear both the short-term and long-term counters without changing the clear interval.

The following figure shows two CLRS commands, one with an argument and the other without.

**FIGURE 14. CLRSnn and CLRS Commands Output**

```
-IDEG   >> Elapsed time= 32 SEC, #samples(short)=63, #samples(long)=63 <<
>CLRS05  >> SHORT-INTERVAL COUNTERS WILL BE RESET EVERY 5 MINUTES <<
>CLRS    >> BOTH SHORT AND LONG COUNTERS HAVE BEEN RESET <<
```

The CLRS? command displays the existing short-term clear interval. If you do not set this option via the CLRS command, Bottleneck Analysis uses the default in the RKNPAR(KOIGBL) OMEGAMON II load module you use. (This is CLRS05 in the TKNPAR(KOIGBLmp) module Candle supplies.)

In the Menu Interface:

1. Set the interval for clearing the short-term data counters by selecting option C, OPTIONS, from the Bottleneck Analysis menu.
2. Remove the > character from before the CLRS command.
3. Add a numeric argument and press Enter.
   The panel updates as shown in the following figure.

**FIGURE 15. Using CLRS from the Menu Interface**

```
> Help PF1               Back PF3               Up PF7                Down PF8
===============================================================================
> To change the value of an option, enter the new value directly after the
> option. For the THRS option, enter a space followed by the new value.

-IDEG   >> Elapsed time= 43 SEC, #samples(short)=61, #samples(long)=61 <<
>bmpxOFF >> BMP ACTIVITY IS BEINGRecorded BY DEXAN <<
>clrl   30  >> long-interval counters will be reset every 30 minutes <<
>clrs   5   >> short-interval counters will be reset every 5 minutes <<
>dbsw   >> I/O Analysis will be done by individual database <<
>stim   >> Sample Time Interval=.5 Seconds <<
>thrs   >> 0 is PDEX wait percent threshold <<
===============================================================================
```
Database sampling (DBSW)

The DBSW minor command controls whether the Bottleneck Analysis collector accumulates bottleneck analysis information by individual database name (DBSWON), or not (DBSWOFF).

Candle provides this option because Bottleneck Analysis must allocate counters for each database for each PSB group you define. The amount of virtual storage Bottleneck Analysis requires can therefore be sizable when the product of the number of PSB groups and databases is large.

The number of bytes Bottleneck Analysis requires is:

$$8 \times (<\text{number of databases}> \times (<\text{MAXG}>+1))$$

MAXG is the maximum number of PSB groups for which Bottleneck Analysis allocated space. For more information about the MAXG command, see the OMEGAMON II for DBCTL Realtime Commands Reference Manual.

You cannot change the database sampling option while the Bottleneck Analysis collector is active. If you attempt to do so, Bottleneck Analysis displays an error message. To change the database sampling option, issue the IDEG and END commands to stop the collector, change the DBSW setting, and then issue the IDEG and BEGN commands to restart the collector.

The following figure shows this series of commands.

**FIGURE 16. Changing the Database Sampling Option (DBSW)**

```
IDEG   >> Collector not active <<
>END    >> Collector now dormant <<
>DBSWOF >> I/O Analysis will not be done by individual database <<
>DBSWON >> I/O Analysis will be done by individual database <<
IDEG   >> Collector not active <<
>BEGN   >> Collector now active <<
```

To display the current database selection option, enter either DBSW? or DBSW (without operands). If you enter ON or OFF, the DBSW command comments itself out after execution by setting its label field to >.

If you do not set this option using the DBSW command, Bottleneck Analysis uses the default in the RKAENV(KOIGBL) OMEGAMON II load module you use. (This default is DBSWON in the TKAENV(KOIGBLmp) module Candle supplies.)

In the Menu Interface, you can control whether the collector collects bottleneck analysis information by following these steps:

1. End data collection as described in “Ending data collection (END)” on page 30.
2. Select option C, OPTIONS, from the Bottleneck Analysis menu.
3. Remove the > character from before the DBSW command and press Enter.
Specifying Collector Options

The panel updates as shown in Figure 17 on page 38.

**FIGURE 17. Using DBSW from the Menu Interface**

<table>
<thead>
<tr>
<th>KDIOPT</th>
<th>VTM</th>
<th>DBCTL</th>
<th>SW</th>
<th>151A</th>
<th>01/02/01</th>
<th>0B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9:20:19</td>
<td></td>
</tr>
</tbody>
</table>

> Help PF1 Back PF3 Up PF7 Down PF8

> Eligible Performance Groups and Other Options

> To change the value of an option, enter the new value directly after the option. For the THRS option, enter a space followed by the new value.

<table>
<thead>
<tr>
<th>IDEG</th>
<th>Elapsed time= 43 SEC, #samples(short)=61, #samples(long)=61</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmpxOFF</td>
<td>BMP ACTIVITY IS BEING RECORDED BY DEXAN</td>
</tr>
<tr>
<td>clrl 30</td>
<td>long-interval counters will be reset every 30 minutes</td>
</tr>
<tr>
<td>clrs 5</td>
<td>short-interval counters will be reset every 5 minutes</td>
</tr>
<tr>
<td>dbsw</td>
<td>I/O Analysis will be done by individual database</td>
</tr>
<tr>
<td>stim</td>
<td>Sample Time Interval= .5 Seconds</td>
</tr>
<tr>
<td>thrs</td>
<td>0 is PDEX wait percent threshold</td>
</tr>
</tbody>
</table>

4. Start data collection as described in “Beginning data collection (BEGN)” on page 29.

**Suspending the collector (SUSP)**

The SUSP command tells the Bottleneck Analysis collector to stop sampling temporarily, but not to clear its short- and long-term buckets. You can use this command to freeze the collector data long enough for you to analyze the information using the MDEX or PDEX command.

The following figure shows the command output for SUSP.

**FIGURE 18. SUSP Command Output**

-IDEG  >> Elapsed time= 3:15 MN, #samples(short)=352, #samples(long)=352

>susp  >> Collector now suspended

Once you finish examining the data, you can start the collector again using the RESM command.

**Resuming collector sampling (RESM)**

The RESM command tells the Bottleneck Analysis collector to resume its sampling. Use the RESM command only after you use the SUSP command to suspend the collector. RESM does not clear any counters, and the data displayed should not be taken as representative of the interval during which collection was suspended.

The following figure shows the command output for RESM.
Specifying Collector Options

FIGURE 19. RESM Command Output

Sampling interval control (STIM)

The Bottleneck Analysis collector analyzes the IMS environment at a fixed sample rate. The STIM command controls the sample time interval that the Bottleneck Analysis collector uses. If you do not supply a STIM value, Bottleneck Analysis uses the default value in the KANPAR(KOIGBL) OMEGAMON II load module. As Candle ships Bottleneck Analysis, this default is 0.5 seconds. You can use the STIM command to vary this rate from a low of 0.1 seconds to a high of 9.9 seconds. STIM interprets the value you enter in tenths of a second (for example, STIM15 indicates a sample time of 1.5 seconds). You should use a STIM value which is small enough to collect at least 30 samples during the short-term interval so your data will be statistically significant.

If you enter the STIM command without a numeric argument, it displays the current sample time. When you enter the STIM command with a valid numeric argument, Bottleneck Analysis clears both the short- and long-term data areas before the collector begins to sample at the new rate. If you change the sample interval, the STIM command comments itself out (by placing a > in its label field) after it executes.

FIGURE 20. STIM Command Output

In the Menu Interface:

1. Set the sample interval by selecting option C, OPTIONS, from the Bottleneck Analysis menu.
2. Remove the > character from before the STIM command.
3. Add a numeric argument and press Enter.

The panel updates as shown in the following figure.
Displaying threshold for MDEX command (MTHR)

Bottleneck Analysis recognizes more IMS execution states than the MDEX command can display on any model 3270 screen. (For more information about the MDEX command, see “Bottleneck Analysis Displays (MDEX/PDEX)” on page 43.) To reduce the size of the display, the MTHR command lets you specify a threshold value. MDEX does not display any state whose short- and long-term average counts are both less than the value specified using MTHR.

The format of this command is MTHR nnnn where nnnn is a value between 0-9999 (9999 actually represents a threshold value of 999.9). If you omit nnnn, MTHR displays the current threshold value.

The following figure shows typical output for the MTHR command.

The default for MDEX is 0. It displays all non-zero execution states. There is no way to force MDEX to display all possible execution states. You can change the default permanently on the $OIDEKAN macro in the RKANPAR(KOIGBL) OMEGAMON II load module. Refer to the Configuration and Customization Guide for details.
Scaling factor for MDEX command (SCAL)

The MDEX command calculates average counts for each of the various execution states to only one decimal place (0.1), so low-volume systems could experience many counts which round down to zero. The SCAL command lets you specify a scaling factor (multiplier), which is multiplied against the value prior to computing the average. Thus you can see where threads are spending time in finer detail. The format of this command is SCAL nnn where nnn is a value between 1-100. If you do not supply a value, Bottleneck Analysis uses the default in the RKANPAR(KOIGBL) OMEGAMON II load module. The default is 1 in the TKANPAR(KOIGBLmp) module Candle supplies.

For example, if you specify a scaling factor of 10, an average count of .03 displays as .3 instead of being rounded down to .0. The following figure shows the SCAL command in combination with the MTHR command.

**FIGURE 23. MTHR and SCAL Commands Output**

<table>
<thead>
<tr>
<th>IDEG</th>
<th>Elapsed time= 2:49 MN, #samples(short)=372, #samples(long)=372 &lt;&lt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTHR</td>
<td>.0 is MDEX display threshold &lt;&lt;</td>
</tr>
<tr>
<td>SCAL</td>
<td>10 is MDEX display scale value &lt;&lt;</td>
</tr>
</tbody>
</table>

Displaying threshold for PDEX command (THRS)

Bottleneck Analysis can detect and display a large number of execution states, so the PDEX display can become quite large. It is very possible, in fact, for PDEX to fill even a 43-line screen with reasons that account for only a tiny percentage of thread wait time. (For more information about the PDEX command, see “Bottleneck Analysis Displays (MDEX/PDEX)” on page 43.)

The THRS minor command sets a threshold which suppresses insignificant states on the PDEX display. If the percentage of total execution time of a particular state (in both the short- and long-term interval) is less than the THRS threshold, PDEX does not include it in the display. This will probably cause the percentages that do display to total less than 100%. The following figure shows a typical THRS command.

**FIGURE 24. THRS Command Output**

<table>
<thead>
<tr>
<th>IDEG</th>
<th>Elapsed time= 4:19 MN, #samples(short)=458, #samples(long)=458 &lt;&lt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>THRS</td>
<td>0 is PDEX wait percent threshold &lt;&lt;</td>
</tr>
</tbody>
</table>

The THRS command accepts values between 0 and 99 percent. If you specify 0 (the default), PDEX does not display execution states which were never seen, since this would generate a large, useless display. Yet, some statistics may still display as zero, because they were rounded down to zero. PDEX only displays true zero statistics if they are for the short interval and the execution state in the long interval is non-zero.
Specifying Collector Options

In the Menu Interface:

1. Set the PDEX wait threshold by selecting option **C, OPTIONS**, from the Bottleneck Analysis menu.

2. Remove the > character from before the THRS command.

3. Add a numeric argument and press Enter.
   The panel updates as shown in the following figure.

**FIGURE 25. Using THRS from the Menu Interface**

Collector abend display (ABCD)

If the Bottleneck Analysis collector abends, Bottleneck Analysis saves PSW and register information. The ABCD command displays this information so you can provide it to Candle Customer Support.

A warning message appears if the Bottleneck Analysis collector abends when you issue the IDEG major command. The design of the collector allows it to recover from the kinds of errors a monitor like Bottleneck Analysis normally encounters (for example, 0C4 or loops because a chain moved), so you should definitely report any abends.

The following figure shows typical ABCD command output when there are no abends to report.

**FIGURE 26. ABCD Command Output**
Chapter overview

This chapter provides information about the MDEX and PDEX display commands.

Chapter contents

Displaying Bottleneck Analysis Data ............................................. 44
Examples of MDEX and PDEX Output ........................................... 46
Factors That Influence the Displays ............................................. 48
Displaying Bottleneck Analysis Data

Introduction

Once the Bottleneck Analysis collector starts and is gathering samples of the activity of threads running in your IMS system, you can display the data it gathers with one or both of the following IDEG minor commands:

**MDEX**
Displays average counts of threads in each of the execution states the Bottleneck Analysis collector monitors.

**PDEX**
Displays the percent of time an average thread spent in each of the execution states the Bottleneck Analysis collector monitors.

Both MDEX and PDEX accept arguments which allow you to include all threads in the display or restrict the display to a single PSB group. The Realtime Commands Reference Manual discusses the concept of PSB groups and how you define them.

Of the two forms of displays, users tend to favor PDEX because they find it presents the data in a form which is usually easier to use. You should realize, however, that the data MDEX and PDEX present is really the same. Given either display and a calculator, you can produce the other within a few minutes.

Bottleneck Analysis Display Commands Format

The following figure shows the format of the Bottleneck Analysis display commands MDEX and PDEX.

**FIGURE 27. MDEX/PDEX Commands Format**

The command label field lets you limit the scope of the Bottleneck Analysis display to specific execution state groups. Here are the valid command label specifications:

- **blank** Displays all execution state analyses.
- **D** Displays database I/O waits only.
Displaying Bottleneck Analysis Data

I Displays IMS internal waits only.
M Displays MVS waits only.

See “Bottleneck Analysis Execution States” on page 51 for an explanation of these groups.

Use the command argument field or the GRP= keyword to limit the scope of the Bottleneck Analysis display. To see only certain PSBs, enter a Bottleneck Analysis group number from 1 to 30 (or the current MAXG value) with the MDEX or PDEX command. For more information on PSB groups or the MAXG command, refer to the Realtime Commands Reference Manual.

If you leave both the command argument field and the GRP= keyword blank, Bottleneck Analysis displays the execution states of all IMS threads.
Examples of MDEX and PDEX Output

Introduction

At first glance, the displays MDEX and PDEX produce look very much alike. Both displays show activity for the execution states listed down the left-hand side over both a short-term interval and a long-term interval. The short-term interval is simply the most recent portion of the long-term interval.

Example of MDEX output

The following figure shows a typical MDEX display.

FIGURE 28. MDEX Command Output

The figure above shows the following information:

- The command specified is MDEX, which displays the average count of threads in each execution state.
- The label field does not contain a selection character, so all types of execution states are eligible for display.
- The argument field does not contain a PSB group number and the GRP= keyword is not specified, so MDEX displays data about all threads.
- Bottleneck Analysis last cleared the long-term counters 1 minute and 41 seconds before MDEX generated this display (Elapsed time= 1:41 MN).
The display threshold set by MTHR is .0, so all non-zero execution states display.

The scaling factor set by SCAL is 1, so MDEX multiples all values that appear by 1, and they are thus true values.

**Example of PDEX output**

The following figure shows a typical PDEX display.

**FIGURE 29. PDEX Command Output**

<table>
<thead>
<tr>
<th>PDEX</th>
<th>Short Term %</th>
<th>Long Term %</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ (Elapsed Time= 1:41 MN)</td>
<td>% 0 50 100</td>
<td>% 0 50 100</td>
</tr>
<tr>
<td>+ Using CPU:</td>
<td>46.0%</td>
<td>44.0%</td>
</tr>
<tr>
<td>+ Using CPU In APPL</td>
<td>(20.0)%</td>
<td>(12.0)%</td>
</tr>
<tr>
<td>+ Using CPU In IMS</td>
<td>(26.0)%</td>
<td>(32.0)%</td>
</tr>
<tr>
<td>+ Scheduling Waits:</td>
<td>3.0%</td>
<td>3.0%</td>
</tr>
<tr>
<td>+ Wait for GU</td>
<td>(1.0)%</td>
<td>(1.0)%</td>
</tr>
<tr>
<td>+ Wait for MPP</td>
<td>(1.0)%</td>
<td>(1.0)%</td>
</tr>
<tr>
<td>+ Wait for Resched</td>
<td>(1.0)%</td>
<td>(1.0)%</td>
</tr>
<tr>
<td>+ Database I/O Wait</td>
<td>36.0%</td>
<td>38.0%</td>
</tr>
<tr>
<td>+ DI21PART</td>
<td>(20.0)%</td>
<td>(24.0)%</td>
</tr>
<tr>
<td>+ BE3PART</td>
<td>(16.0)%</td>
<td>(14.0)%</td>
</tr>
<tr>
<td>+ IMS Activity</td>
<td>15.0%</td>
<td>8.0%</td>
</tr>
<tr>
<td>+ ISWITCHED to CTL</td>
<td>(7.0)%</td>
<td>(2.0)%</td>
</tr>
<tr>
<td>+ IRLM Conflict Wait</td>
<td>(8.0)%</td>
<td>(6.0)%</td>
</tr>
<tr>
<td>+ Avg. Trans Executing</td>
<td>5.4%</td>
<td>3.2%</td>
</tr>
</tbody>
</table>

The figure above shows the following information.

- The command specified is PDEX, which displays the percentage of time an average thread was in each execution state.
- The label field does not contain a selection character, so all types of execution states are eligible for display.
- The argument field does not contain a PSB group number and the GRP= keyword is not specified, so data about all threads displays.
- Bottleneck Analysis last cleared the long counters 1 minute and 41 seconds before PDEX generated this display (**Elapsed time= 1:41 MN**).
Factors That Influence the Displays

List of factors

A number of factors determine what data the MDEX and PDEX commands display. These factors include:

- Whether or not the collector gathered data about message-driven BMPs. The BMPX command controls the collection of this data. “BMP Thread Sampling (BMPX)” on page 33 describes the BMPX command.

- Whether or not the collector gathered data about DL/I I/O for each individual database, or just for database I/O as a whole. The DBSW command controls the collection of this data. “Database sampling (DBSW)” on page 37 describes the DBSW command.

- Whether the MDEX or PDEX command restricts its display to a certain class of execution states. To indicate the class of execution states, put one of a set of characters in the label field (column 1) of the command as “Bottleneck Analysis Display Commands Format” on page 44 describes.

- Whether the MDEX or PDEX command restricts its display to a single PSB group. A numeric argument from 01 to 30 following the command indicates that MDEX or PDEX should only display data for PSBs in a certain (user defined) group. For example, PDEX02 causes bottleneck analysis to only display data for PSBs which the collector determined to fall in Group 2. Bottleneck Analysis groups can be defined dynamically for realtime analysis using the SETG command. The Realtime Commands Reference Manual explains how to define groups.

- For MDEX, the current display threshold and scaling factor. Only execution states whose short- and long-term average counts are less than the value specified using the MTHR command display. “Displaying threshold for MDEX command (MTHR)” on page 40 explains how to vary the display threshold. The SCAL command lets you further control the display by letting you set a scaling factor. With it, you can display execution states whose short- and long-term average counts might ordinarily be too small to see. How you specify the scaling factor is explained in “Scaling factor for MDEX command (SCAL)” on page 41.

- For PDEX, the current display threshold. PDEX only displays execution states which account for more than the percentage of execution time specified using the THRS minor command. If you specified a percentage greater than 0 using THRS, then the execution state numbers may not add up to 100%. On the other hand, setting the percentage greater than zero reduces the size of the display. “Displaying threshold for PDEX command (THRS)” on page 41 describes the THRS command.
Factors That Influence the Displays

- The actual activity on your system has the greatest effect on the MDEX and PDEX displays. MDEX and PDEX will not display an execution state if there were no threads seen in it. It would be impractical to do otherwise since there are so many possible execution states in IMS.

- Bottleneck Analysis displays execution states even for well-tuned systems with good response time. Unless response time is slow, these states do not necessarily indicate problems.
Factors That Influence the Displays
Chapter overview

This chapter provides information about the Bottleneck Analysis execution states.

Chapter contents

About Execution States ............................................. 52
CPU Usage .............................................................. 53
Database I/O Waits .................................................... 55
MVS Waits ............................................................. 56
IMS Waits ............................................................. 61
About Execution States

Introduction

The MDEX and PDEX displays group the Bottleneck Analysis execution states. The first line of the display is a subtotal covering the entire PSB group. The individual execution states comprising that subtotal follow, subject to the threshold specified using the MTHR minor command for MDEX, or the threshold percentage specified using the THRS minor command for PDEX.

Execution state groups

The subtotals by group are not enclosed in parentheses, while the numbers for the individual execution states are, to visually distinguish them. The execution state groups are as follows:

- CPU Usage
- Database I/O Waits
- MVS Waits
- IMS Waits

The following sections discuss these groups and the individual execution states which comprise them.
CPU Usage

Introduction

Bottleneck analysis as a methodology focuses on CPU usage. Removing bottlenecks enables threads to use more CPU per unit of time and complete faster. The individual execution states in the CPU usage group describe how much CPU the threads use and where (for example, CPU used in DL/I versus CPU used in the application program itself). All of these states are executing states—they are only experienced by threads actually running in the control region.

Note: Because of their importance, Bottleneck Analysis always displays the CPU Usage execution states, even when you use a label field character to restrict the display to a certain class of execution groups as described in “Bottleneck Analysis Display Commands Format” on page 44.

Using CPU states

You can use the Using CPU states to infer activity which Bottleneck Analysis cannot measure directly. For example, a shift away from Using CPU toward Database I/O, in the absence of increased I/O contention, indicates that the databases being accessed may require reorganization. An abrupt, massive increase in Using CPU indicates a possible CPU hardware problem. For example, the cache buffer or the TLB may have gone out of service, and the message issued by the machine check handler may have gone unnoticed or been misunderstood.

Using CPU in APPL

The collector includes a thread in the Using CPU in APPL state when it finds the thread is actually executing instructions on a CPU. The program had no DL/I function in progress, so the collector ascribes its use of the CPU to application processing.

In a normal IMS system, this number will be small, even tiny, compared to the other execution states. However, this is one of the most important statistics you can watch because:

- Almost every tuning trick increases this number (and decreases the total of the other execution states by the same amount). Indeed, a thread executing 100% in this state is not degraded at all in the Bottleneck Analysis sense of the term unless it is looping.
- Almost every performance problem decreases this number (and increases the total of the other execution states by the same amount). For example, if I/O contention slows down an application’s access to a database, its execution profile shifts towards Database I/O and away from this execution state.
Using CPU in IMS

The collector found the thread actually executing instructions on a CPU. The program had a DL/I call in progress, so the collector ascribed its use of the CPU to IMS processing. An example of such processing is the CPU used by DL/I action modules to search a buffer pool for a record before issuing an I/O to retrieve it.
Database I/O Waits

Introduction

Many database DL/I requests can be satisfied from records already in a buffer pool. Others, however, require that IMS perform physical I/O. MDEX and PDEX show a subtotal for database I/O. Then, if you specified that such statistics be kept by the collector, it shows the breakdown of the I/O by database. No further breakdown, such as by dataset group or individual dataset, is available from Bottleneck Analysis.

The individual execution states within this category are the DMB names of each database to which a thread did I/O.

I/Os to a secondary index

A wait reason of INDEXPCB indicates I/O activity caused by secondary index maintenance. When maintenance is performed on a secondary index, IMS constructs a PCB to that index. Although you may not be processing in secondary sequence, I/Os to a secondary index may be caused by:

- Designing your database so that the segment or fields used as the key in secondary sequence are updated heavily by other applications.
- Changes to the primary database. Even if the application program is not using the secondary index, maintenance must be performed on the index, and a PCB constructed for it, when the primary database is changed.

Also, there is no way to tell from the Database I/O Waits statistic alone which stages of I/O processing (for example, active on the UCB or waiting in the logical channel queue) the I/O operations for the databases were in. A tool such as Bottleneck Analysis can give you some view of this.

In addition to IMS/VSE full-function databases, physical I/O operations performed to data entry databases (DEDBs) are also reported. If statistics by database are requested, the area name is shown as the execution state.
Introduction

The process of executing a thread or a batch message processing region can be blocked by contention for resources managed by MVS (rather than by IMS), such as the CPU. The individual execution states in this group show you when and where threads get slowed down by resources managed by MVS.

CPU Wait (CTL)

The collector found the thread ready to run on a CPU within the IMS control region, but all online CPUs in the complex were running other work. The CPUs might be running other IMS control region tasks, IMS dependent regions, or memories not related to IMS, such as TSO.

Note: The collector does not count itself in this analysis. If the only reason a thread is not executing on a CPU is that the collector is executing, the collector counts the thread as Using CPU. This is a standard sampling technique for measuring CPU usage by workload.

Also, the collector cannot see CPU Waits resulting from work executing with a higher dispatching priority than the collector. For example, the collector cannot see CPU Waits resulting from disabled global SRB, higher priority address spaces such as *MASTER*, and local SRBs in the IMS CTL region. If there is much of this higher priority activity, it will skew the measurements. Some amount of CPU Wait—and perhaps even other states such as Database I/O Wait—are seen as Using CPU.

CPU Wait (DEP)

The collector found the thread ready to run on a CPU within its region, but all online CPUs in the complex were running some other MVS address space. The CPUs might be running the IMS control region, other dependent regions, or address spaces not related to IMS, such as TSO.

Swapping In

The collector found the thread active inside a CICS™ region which was in the process of being swapped in by MVS.

CTL Private Page

The collector found the thread waiting for one of the following reasons:

- Its ITASK took a page fault for a page in the private area of the control region’s address space.
- Its ITASK was ready to use the CPU, but it was waiting in the IMS dispatcher’s READY queue while the control task was serialized by a page fault taken in the private area by some other ITASK.
The reason why the IMS control region can stand so little paging is that when one ITASK page faults in the CTL region, all ITASKs running in the CTL region halt.

**CTL EXT Priv Page**

This execution state is the same as the CTL Private Page execution state, except that the private page MVS referenced is located in the control region’s extended private area, which is located above the 16 megabyte addressing line.

**DEP Private Page**

The thread was waiting for the resolution of a page fault for a page within its private area.

**DEP EXT Priv Page**

This execution state is the same as the DEP Private Page execution state, except that the private page MVS referenced is located in the region’s extended private area, above the 16 megabyte addressing line.

**LPA Page**

The collector found the thread’s execution blocked for one of two reasons:

- The application program (most likely compiler runtime routines copied to LPA or some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in the pageable link pack area (PLPA).
- The application program processing the thread had ISWITCHed to the IMS control region. While there, the collector found its ITASK waiting for the resolution of a page fault in LPA or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in LPA.

**ELPA Page**

The collector found the thread’s execution blocked for one of two reasons:

- The application program (most likely compiler runtime routines copied to LPA or some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in the extended pageable link pack area (EPLPA).
- The application program processing the thread had ISWITCHed to the IMS control region. While there, the collector found its ITASK waiting for the resolution of a page fault in extended LPA, or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in extended LPA.
Cross Memory Page

The collector found the thread active inside a CCTL. The IMS system was running with LSO=X. The application program processing the thread had issued a DL/I call which was being processed by IMS modules running in the control region, but under the ASCB/TCB of the thread through the MVS Cross Memory Services. DL/I processing had taken a page fault for a page in the control region’s private area.

In this situation, the thread’s execution is suspended while the page fault is resolved. Processing by the IMS control task is unaffected.

CTL Private Page (GFA)

The collector found the thread active. The application program processing the thread was ISWITCHed to the IMS control region, and one of two things happened:

- Its ITASK took a page fault for a page in the private area of the control region’s address space.
- Its ITASK was ready to use the CPU, but it was waiting in the IMS dispatcher’s READY queue while the control task was serialized by a page fault taken in the private area by some other ITASK.

In either case, the page fault occurred at a time when the Available Frame Queue (AFQ) was empty, so the request for the resolution of the page fault had to be put on a special Real Storage Manager (RSM) queue, called the General Frame Allocation (GFA) queue, while the System Resources Manager (SRM) stole frames to replenish the AFQ.

GFA wait is not a major problem, because SRM STEAL is not the only source of replenishment for the AFQ. Every time an address space is physically swapped out, the frames it occupied go on the AFQ. Also, when a FREEMAIN is issued, all frames backing the virtual storage released are made available.

DEP Private Page (GFA)

The collector found the thread active. The application program processing the thread (or possibly some IMS module running as a subroutine of the application program) was waiting for the resolution of a page fault for a page in its private area. The request for resolution of the page fault was found on the general frame allocation (GFA) queue, indicating that it had occurred when RSM was out of frames (AFQ of 0).

Refer to the discussion of GFA under CTL Private Page (GFA).
CSA Page (GFA)

The collector found the thread active with its execution blocked for one of two reasons:

- The application program—or, more likely, some IMS module running as a subroutine of the application program—was waiting for the resolution of a page fault for a page in the common service area (CSA).
- The thread had ISWITCHed to the IMS control region. While there, the collector found its ITASK waiting for the resolution of a page fault it took in CSA, or waiting to use the CPU behind another ITASK which had serialized the CTL Task by taking a page fault in CSA.

In either case, the page fault occurred at a time when the available frame queue (AFQ) was empty, so the request for the resolution of the page fault had to be put on the general frame allocation (GFA) queue, and that is where the collector found it.

Refer to the discussion of GFA under CTL Private Page (GFA).

LPA Page (GFA)

The collector found the thread with its execution blocked for one of two reasons:

- The application program—or, more likely, some IMS module running as a subroutine of the application program—was waiting for the resolution of a page fault for a page in the pageable link pack area (PLPA).
- The thread had ISWITCHed to the IMS control region. While there, the collector found its ITASK waiting for the resolution of a page fault it took in LPA, or waiting to use the CPU behind another ITASK which had serialized the CTL task by taking a page fault in LPA.

In either case, the page fault occurred at a time when the available frame queue (AFQ) was empty, so the request for the resolution of the page fault had to be put in the general frame allocation (GFA) queue, and that is where the collector found it.

Refer to the discussion of GFA under “CTL Private Page (GFA)” on page 58.
X-MEM Page (GFA)

The collector found the thread active. The IMS system was running with LSO=X or LSO=S. The thread issued a DL/I call, which was being processed by IMS modules running in the control region or DLISAS, but under the ASCB/TCB of the thread using MVS Cross Memory Services. DL/I processing took a page fault for a page in the control region’s private area. The collector found the request for resolution of the page fault on the GFA queue, indicating that it occurred when the AFQ was empty.

In this situation, the thread’s execution is suspended while the page fault is resolved. Processing by the IMS control task is unaffected.

Refer to the discussion of GFA under “CTL Private Page (GFA)” on page 58.
IMS Waits

Introduction

While executing in a CICS region or a batch message processing region, a thread can be blocked by contention for resources managed by IMS (rather than by MVS), such as an IMS latch. The individual execution states in this group show you when and where threads are slowed down by resources managed by IMS.

ACTL Latch

The collector found the thread waiting for the ACTL (statistics logging) latch. IMS uses the ACTL latch to serialize access to the DC Monitor log buffers and control blocks. When the DC Monitor is active, the control region and all parallel DL/I tasks compete for this latch.

ADSC Directory Latch

The collector found the thread waiting for the ADSC directory latch.
The ADSC directory latch is used to serialize access to the ADSC directory. This is a wait built on top of the dynamic control block (CBTS) latch. The ADSC directory latch is obtained in the open and close data entry database (DEDB) modules, and Fast Path command processing modules.

AUTH Latch

The collector found the thread waiting for the AUTH latch.
The AUTH latch is the authorized processing latch.

CBTS Latch

The collector found the thread waiting for the CBTS latch. IMS uses the CBTS latch to serialize alterations to dynamic control block chains (IPAGES) within dynamic storage management.

Conversation Checkpoint Latch (Executing)

The collector found the thread waiting for the Conversation Checkpoint latch.

DBBP Latch

The collector found the thread waiting for the DBBP latch. The DBBP latch serializes accesses to database buffers and their associated control blocks.

DB System Checkpoint Latch (Executing)

The collector found the thread waiting for the DB System Checkpoint latch.
DC System Checkpoint Latch (Executing)
The collector found the thread waiting for the DC System Checkpoint latch.

DC Terminal Latch (Executing)
The collector found the thread waiting for the DC Terminal latch.

DC User Latch (Executing)
The collector found the thread waiting for the DC User latch.

DDIR Pool Latch (Executing)
The collector found the thread waiting for the DDIR Pool latch.

DEDB Area Lock
The collector found the thread waiting for the DEDB area lock.
The DEDB area lock serializes updates to a DEDB area dataset. There is a
different lock for each area. The DEDB area lock is obtained in sync point
processing.
A DEDB area lock is always taken before the DMAC latch to avoid the
possibility of a deadlock.

DEDB Segment
The collector found the thread waiting to obtain control of a resource in a data
entry database (DEDB). The resource under control is a unit of work (UOW),
and it is represented by an XCRB control block.

DMAC Latch
The collector found the thread waiting for the DMAC latch.
The DMAC latch is used to serialize updates to DEDB area datasets. Sync
point processing modules require the DMAC latch. Updates are recorded in
the log records. The physical update of the DEDB area does not take place
until the updates are written to the IMS log.

DMAC Share Latch
The collector found the thread waiting for the DMAC share latch.
The DMAC share latch is used to serialize access to DMAC control blocks.
Open and close area dataset processing modules require the DMAC share
latch in exclusive mode.
**DMBE Latch**

The collector found the thread waiting for the DMBE latch. The DMBE latch is used to serialize the dynamic insertion and removal of control blocks associated with databases (DMBs).

**Fast Path Buffer**

The collector found the thread waiting for a Fast Path buffer. Buffer allocation for an IMS Fast Path Region (IFP) is under the maximum number of allowable buffers specified by the normal buffer allocation (NBA) execution parameter.

**Fast Path IWAIT in Term**

The collector found the Fast Path application program processing the thread IWAITing while the application program was terminating.

**Fast Path Other Abend**

The collector found the Fast Path application program processing the thread IWAITing and that the application program was terminating due to an abend.

**Fast Path Overflow Buffer Allocation**

The collector found the application program processing the thread waiting for a Fast Path overflow buffer. The thread has reached the maximum number of buffers allowed, (as specified by the NBA execution parameter) and overflow processing is in progress. Overflow processing increased the buffer allocation by the amount allowed for overflow (OBA).

**Fast Path Overflow Buffer Lock**

The collector found the application program processing the thread waiting for the Fast Path overflow buffer lock.

If a thread attempts to allocate buffers beyond the normal number of buffers allowed (NBA), overflow processing takes place. Overflow processing attempts to increase the buffer allocation by the amount allowed for overflow (OBA). The overflow lock is required to allocate more buffers than the number specified by normal buffer allocation. The overflow lock is enqueued by the thread and held until sync point processing, at which time the buffer allocation is returned to normal.

**Fast Path Pseudo Abend**

The collector found the Fast Path application program processing the thread IWAITing while the thread was pseudo abended.
Fast Path Resource Latch

The collector found the application program processing the thread waiting for the Fast Path resource latch.

The Fast Path Resource latch serializes access to the exclusive control resource blocks (XCRBs). XCRBs represent the status of a currently-owned (and possibly exclusively-controlled) resource within a DEDB area.

Fast Path Syncpoint Processing (Executing)

The collector found the thread in sync point processing. During this state, the dependent region is using CPU; however, the amount of time spent in this state should be small.

Fast Path Sync Lock

The collector found the application program processing the thread waiting for the Fast Path sync lock.

The Fast Path sync lock serializes the sync point processing in threads with check-point processing and other activities which stop a DEDB area, such as issuing a /STOP AREA, /STOP ADS, or /DBR AREA command, or if a physical I/O error is detected.

The sync point processing function requests the Fast Path sync lock in share mode. All other activities require the lock to be held in exclusive mode.

FNCB Latch

The collector found the application program processing the thread waiting for the FNCB latch.

The FNCB latch serializes access to the notify control block chain. The notify facility is used for communication by various functions like opening or closing a DEDB, and IMS commands.

GCMD Latch

The collector found the thread waiting for the GCMD latch.

The GCMD latch is the global command latch.

GEN1 Latch

The collector found the thread waiting for a generic latch.

The generic latch is a class of latches which currently consists of the DMBE (data management block) and DBBP (database buffer pool) latches. The latches are generic in that there is one DMBE latch for each DMB defined to the IMS system, and one DBBP latch for each subpool used by the ISAM/OSAM buffer handler.
HDSM Latch
The collector found the thread waiting for the HDSM latch. The HDSM latch serializes processing of DL/I HD space management operations.

IRLM Conflict Wait
The collector found the thread waiting for a response to a request the thread issued to the IRLM address space.
This wait reason can appear when there is a database conflict or when IRLM is waiting for internal IRLM processing.

ISWITCHed to CTL
The collector found that the thread executed an ISWITCH macro. The thread is still executing, but in the IMS control region instead of in the CICS region.
This wait reason occurs whenever a thread needs the services of the DBCTL region. When this happens, the thread gets ISWITCHed to CTL and waits to request the service it needs from the control region TCB. The wait is longer if the TCB itself is waiting for control of the CPU (for example, if the control region has a high paging rate).
There are a number of conditions under which the thread is executing in the control region. These include, but are not limited to:
- I/O to ISAM files
- EOV processing on all database datasets (when required to free up a buffer)
- Users issuing large numbers of IMS commands
- Services Fast Path Wait For Input (WFI) GU calls
- Create and terminate thread processing
- Complete sync-point processing

ISWITCHed to Fast Path TCB
The collector found that the thread executed an ISWITCH macro. The thread is still executing, but in the control region under the Fast Path TCB.

ISWITCHed to LSO
The collector found that the thread executed an ISWITCH macro. The thread is still executing, but in the control region instead of the CICS region. The ISWITCH was done using the Local Storage Option (LSO).
For LSO=Y, instead of running in the control task, the thread is running under a subtask in the control region that is reserved for its use only. LSO=Y costs a substantial amount of CPU, but saves virtual storage in CSA at the cost of virtual storage in the private area of the IMS control region.
**IMS Waits**

**IWAIT in IMS Dispatcher**

IWAIT in IMS dispatcher is a measure of the IMS dispatching queue for CICS region activities. The collector found the thread IWAITing in the IMS dispatcher and could not attribute the condition to any of the DL/I oriented states listed earlier in this chapter.

As the workload rises, this value also rises. This number should be small.

**IWAIT in Term**

The collector found the thread IWAITing. It also found an indication that the application program was terminating. The collector could not attribute the status of the thread to any of the states listed earlier in this chapter.

*Note:* This execution state should be small. Termination IWAITs may be attributed to sync point processing of an update thread.

**LOGL Latch**

The collector found the thread waiting for the LOGL latch.

The IMS logical logger uses the LOGL latch to serialize access to the buffers that the physical logger writes out to the online log dataset (OLDS). If the physical logger gets behind, ITASKs queue on this latch.

**LQB Pool Latch (Executing)**

The collector found the thread waiting for the LQB Pool latch.

**LU 6.2 Manager Latch (Executing)**

The collector found the thread waiting for the LU 6.2 Manager (LUM) latch.

**OBFM Latch**

The collector found the thread waiting for the OBFM latch. The OBFM latch serializes accesses to OSAM buffers and their associated control information.

**Open/Close Latch**

The collector found the thread waiting for the open/close latch.

The open/close latch serializes resources to Fast Path databases.

**Other Abend**

The collector found the thread IWAITing. It also found an indication that the thread was terminating due to an abend.
Other DL/I IWAIT

The collector found the thread executing in DLISAS, but it could not attribute the condition to any of the DL/I-oriented states listed earlier in this chapter.

Other Latch

The collector found the thread in ISERWAIT (latch wait), but not waiting for any of the latches described above.

Other Wait

The collector found a thread waiting, but could not attribute its execution state to any of the categories documented earlier in this chapter.

PI Wait

The collector found the thread IWAITing for a program isolation (PI) resource. Program isolation is the mechanism which allows databases to be simultaneously accessed, and even updated, by different users.

A PI resource is essentially a DMB number to identify the physical database, and an RBA (relative byte address) to identify the physical block being held. When two programs attempt to access the same PI resource at the same time, one must wait.

Pseudo Abend

The collector found the thread IWAITing and terminating due to a pseudo abend.

Queue Manager Latch (Executing)

The collector found the thread waiting for the Queue Manager latch.

SMB Queue Latch (Executing)

The collector found the thread waiting for the SMB Queue latch.

SMGT Latch

The collector found the thread waiting for the SMGT (storage management) latch.

The IMS storage management module (DFSISMN0) uses this latch to serialize access to various control blocks and buffers, such as the message queue, CIOP, CWAP, DMB, and PSB pools.

TCT Block Latch (Executing)

The collector found the thread waiting for the TCT Block latch.
**TM Subqueue Latch (Executing)**

The collector found the thread waiting for the TM Subqueue latch.

**VBFM Latch**

The collector found the thread waiting for the VBFM latch. The VBFM latch serializes accesses to VSAM buffers and their associated control information.

**VSAM Buffer Wait**

The collector found the thread waiting in the DL/I VSAM buffer handler. The buffer handler acquires buffers for VSAM database I/O.

**VTCB Pool Latch (Executing)**

The collector found the thread waiting for the VTCB Pool Latch.

**Wait HSSP PVT Pool**

The collector found the thread executing inside a dependent HSSP BMP region. The thread is waiting for space in the HSSP private pool.

**XCNQ Latch**

The collector found the thread waiting for the XCNQ latch.

Program isolation uses the XCNQ latch to serialize access to the PI ENQ/DEQ queues and control blocks.
Appendix Overview
This appendix summarizes the Bottleneck Analysis commands.

Appendix Contents

- Bottleneck Analysis Commands ........................................ 70
- PSB Group Commands .................................................. 72
Bottleneck Analysis Commands

Introduction

The following sections list each bottleneck analysis command.

Major Command

IDEG

Controls execution of Bottleneck Analysis. A hyphen (-) in column 1 attaches the Bottleneck Analysis collector.

Minor Commands of IDEG

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCD</td>
<td>Data collector ABEND completion code.</td>
</tr>
<tr>
<td>BEGN</td>
<td>Begins Bottleneck Analysis data collector.</td>
</tr>
<tr>
<td>BMPX</td>
<td>Displays current setting of BMP switch.</td>
</tr>
<tr>
<td>BMPXON</td>
<td>Data collector ignores BMP activity.</td>
</tr>
<tr>
<td>BMPXOFF</td>
<td>Data collector does not ignore BMP activity.</td>
</tr>
<tr>
<td>CLRL</td>
<td>Specifies long-term clearing interval (minutes).</td>
</tr>
<tr>
<td>CLRS</td>
<td>Specifies short-term clearing interval (minutes).</td>
</tr>
<tr>
<td>DBSW</td>
<td>Displays current setting of database I/O switch.</td>
</tr>
<tr>
<td>DBSWON</td>
<td>Collects and displays database I/O values by individual database name.</td>
</tr>
<tr>
<td>DBSWOFF</td>
<td>Collects and displays total database I/O values only.</td>
</tr>
<tr>
<td>DTCH</td>
<td>Detaches collector subtask.</td>
</tr>
<tr>
<td>END</td>
<td>Stops data collector.</td>
</tr>
<tr>
<td>MDEX</td>
<td>Displays bottleneck analysis results by count (group nn).</td>
</tr>
<tr>
<td>MTHR</td>
<td>Specifies display threshold for MDEX.</td>
</tr>
<tr>
<td>PDEX</td>
<td>Displays bottleneck analysis results by percentage (group nn).</td>
</tr>
<tr>
<td>RESM</td>
<td>Resumes collector.</td>
</tr>
<tr>
<td>SCAL</td>
<td>Specifies scaling factor for MDEX.</td>
</tr>
<tr>
<td>STIM</td>
<td>Sets collection interval (tenths of a second).</td>
</tr>
<tr>
<td>SUSP</td>
<td>Suspends collector.</td>
</tr>
<tr>
<td>THRS</td>
<td>Suppresses execution states that occur less than nn percent of the time on PDEX display.</td>
</tr>
</tbody>
</table>
To limit the MDEX and PDEX displays to a specific execution state group, enter one of the following arguments in the label field (column 1):

- **blank**: All execution state analyses
- **D**: DB I/O waits
- **M**: MVS waits
PSB Group Commands

Introduction

The following table lists the PSB group commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLBLcc</td>
<td>Displays or changes current suffix of KOIGBLcc module; entering cc causes new module to be loaded.</td>
</tr>
<tr>
<td>MAXGnn</td>
<td>Dynamically controls the number of PSB groups that OMEGAMON II supports.</td>
</tr>
<tr>
<td>aSETGnnccc</td>
<td>Displays or changes contents of a PSB group. In this command, a is the function to be invoked, nn is the group number (enter 99 for all groups), and ccc is the entry specification (PSB=xxx). The functions are:blank adds an entry to group definition.</td>
</tr>
<tr>
<td>A</td>
<td>Adds an entry to group definition.</td>
</tr>
<tr>
<td>C</td>
<td>Creates a group.</td>
</tr>
<tr>
<td>D</td>
<td>Deletes an entry from group.</td>
</tr>
<tr>
<td>L</td>
<td>Lists the contents of group (default).</td>
</tr>
<tr>
<td>X</td>
<td>Deletes all group contents.</td>
</tr>
</tbody>
</table>
Guide to Candle
Customer Support

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 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
- 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 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
- Assigned Support Center Representative (ASCR)
- Maintenance Assessment Services (MAS)
- Multi-Services Manager (MSM)

Customer Support Contact Information . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
- 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.

| Severity 1 | Crisis | A crisis affects your ability to conduct business, and no procedural workaround exists. The system or application may be down. |
| Severity 2 | High   | A high-impact problem indicates significant business effect to you. The program is usable but severely limited. |
| Severity 3 | Moderate | 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. |
| Severity 4 | Low    | A low-impact problem is a “how-to” or an advisory question. |
| Severity 5 | Enhancement Request | 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. |
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></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/.

Select Support Contacts from the list on the left of the page.
Symbols

*MDEX minor 70
*PDEX minor 70
/DBR AREA 64
/STOP
  ADS 64
  AREA 64

A

ABCD 42
ABCD minor 70
abend
  other 66
  termination due to 63
ACTL Latch 61
Adobe portable document format 10
ADSC Directory Latch 61
APPL 53
APPL, using CPU in 53
ASCR
  assigned support center representative 78
aSETGnn 72
assigned support center representative
  ASCR 78
attaching the collector 26–28
  at an MVS console 28
  from OMEGAMON II 26
  when OMEGAMON II initializes 27
AUTH Latch 61
Available Frame queue 58

B

batch processing regions
  ignoring 33
begin data collection 29
BEGN 37
BEGN minor 29, 70
BMP
  transaction sampling (BMPX) 33, 48
BMPX 48
  BMPX minor 70
  from menu interface 34
  minor 34
BMPX? 33
BMPXOFF 33
BMPXOFF minor 70

BMPXON 33
BMPXON minor 70
bottleneck analysis
  bytes required 37
  commands summary 70
  description 8
  invoking 29
  menu 19
  methodology 53
  buckets (virtual storage area) 24

C

CBTS Latch 61
clear interval
  long-term 34
  short-term 34
clearing counters 34–35
CLRL 34
CLRL minor 70
CLRL05 and CLRL
  commands output 35
CLRS 35
CLRS minor 70
CLRS? 36
CLRSnn and CLRS
  command output 36
collector 24–42
  abend 42
  attaching 26
  controlling 29
  detaching 31, 32
  leave system command 30
  operation 24, 25
  options, specifying 33
  restart 37
  starting 27, 29
collector abend display (ABCD) 42
command argument field 45
command label field 44
commands
  major 70
  minor 70
  transaction group 72
controlling
  collector 29
  database sampling 37
Conversation Checkpoint latch (Executing) 61
counters
clearing 35
counters (virtual storage area) 24
CPU
usage 53
using in APPL 53
using in IMS 54
wait (CTL) 56
wait (DEP) 56
Cross Memory Page 58
CSA page
GFA 59
CTL
EXT Priv Page 57
Private Page 56
private page (GFA) 58
customer support
base maintenance plan 74
contact information 80
enhanced support services 78
eSupport 75
severity levels 75
telephone support 74

D

data collection
starting 26, 27, 29
data entry databases (DEDBs) 55
database
I/O Waits execution states group 55
database sampling
changing 37
option 37
database sampling (DBSW) 37
DB System Checkpoint Latch 61
DBBP Latch 61
DBCTL environments 49
DBSW 37, 48
DBSW minor 37, 70
DBSW? 37
DBSWOFF 37
DBSWOFF minor 70
DBSWON 37
DBSWON minor 70
DC Monitor 61
DC system checkpoint Latch 62
DC Terminal Latch 62
DC User Latch 62
DDIR Pool Latch 62
DEDB
area lock 62
segment 62
DEP
EXT Priv Page 57
Private Page 57
private page (GFA) 58
detaching the collector 31, 32
detaching the collector (DTCH) 31
display
limiting 45
display commands format 44
display size
reducing 40
display threshold for MDEX command (MTHR) 40
display threshold for PDEX command (THRS) 41
displays
factors influencing 48
displays (MDEX/PDEX) 44–49
DL/I IWAIT, large 67
DMAC
control blocks 62
latch 62
share latch 62
DMBE latch 63
documentation set information 10
DTCH
from menu interface 32
minor 31
DTCH minor 31, 70
dynamic control block chains (IPAGES) 61

E

ELPA Page 57
END 30, 37
minor 25
END minor 70
Ending Data Collection (END) 30
eSupport
customer support 75
documentation set information 10
exclusive control resource blocks 64
execution states ??–68
ACTL latch 61
ADSC directory latch 61
AUTH latch 61
CBTS latch 61
class 48
CPU
usage group 52, 53
using in APPL 53
using in IMS 54
wait (CTL) 56
wait (DEP) 56
cross memory page 58
CSA page GFA 59
CTL
Index 83

EXT priv page 57
  private page 56
  private page (GFA) 58
database I/O
  operations 55
  waits group 52
DBBP latch 61
DEDB
  area lock 62
  segment 62
DEP
  EXT priv page 57
  private page 57
  private page (GFA) 58
DMAC
  latch 62
  share latch 62
DMBE latch 63
ELPA page 57
fast path
  buffer 63
  IWAIT in term 63
  other abend 63
  overflow buffer allocation 63
  overflow buffer lock 63
  pseudo abend 63
  resource latch 64
  sync latch 64
  sync processing 64
fast path syncpoint processing (executing) 64
FCNB latch 64
GCMD latch 64
GEN1 latch 64
HDSM latch 65
IMS waits group 52
intent conflict (executing) 61, 62, 66, 67, 68
IRLM conflict wait 65
ISWITCHed to
  CTL 65
  fast path TCB 65
  LSO 65
IWAIT in
  IMS dispatcher 66
  term 66
LOGL latch 66
LPA page 57
LPA page (GFA) 59
MVS waits group 52
OBFM latch 66
open/close latch 66
other
  abend 66
  DL/I IWAIT 67
  latch 67
  wait 67
  PI wait 67
  pseudo abend 67
  SMGT latch 67
  swapping in 56
  syncpoint processing 64
  VBFM latch 68
  wait HSSP PVT pool 68
  XCNQ latch 68
  X-MEM page (GFA) 60
  extended pageable link pack area (EPLPA) 57

F
  fast path
    application IWAITing 63
    buffer 63
    IWAIT in term 63
    other abend 63
    overflow buffer allocation 63
    overflow buffer lock 63
    pseudo abend 63
    resource latch 64
    sync latch 64
    syncpoint processing 64
    TCB 65
    thread waiting for buffer 63
    FNCB latch 64
    frame
      allocation 58, 59
      making available 58
    FREEMAIN 58

G
  GCMD latch 64
  GEN1 latch 64
  GFA queue 58
  GLBL major 72
  GLOBAL= parameter 27, 28
  groups
    number supported 24
    PSB 24
  GRP= keyword 45, 48

H
  hardware CPU degradation 53
  HDSM latch 65
  HSSP private pool 68

I
  IDEG 37
=BEGIN keyword 27
from menu interface 26
major 26, 29, 31
with START Bottleneck Analysis deleted 27
IDEG and BEGIN
commands output 29
IDEG and END
commands output 30
IDEG major 26, 70
IDEG minors 70
IMS
resource contentions 61
waits execution states group 61
invoking bottleneck analysis 19, 29
IRLM
conflict wait 65
waiting 65
ISERWAIT (latch wait) 67
ISWITCHed to
  CTL 65
  fast path TCB 65
LSO 65
IWAIT in
  IMS dispatcher 66
  Term 66

K
KOIGBL module 27
  specifying 27
KOIGBLmp 27
KOIGBLmp module 27

L
latch 61–63, 64
  ACTL 61
  ADS directory 61
  AUTH 61
  CBTS 61
  conversation checkpoint 61
  DB system checkpoint 61
  DDBP 61, 64
  DC system checkpoint 62
  DC terminal 62
  DC user 62
  DDIR pool 62
  DEDB area lock 62
definition 66
  DMAC 62
  DMAC share 62
  DMEB 64
  fast path resource 64
  FNCB 64
  GCMD 64
  GENI 64
generic 64
  HDSM 65
  LOGL 66
  LQB pool 66
  LU 6.2 manager 66
  OBFM 66
  open/close 66
  other 67
  queue manager 67
  SMB queue 67
  SMGT 67
  SMTG (storage management) 67
  syncpoint processing 64
  TCT block 67
  TM subqueue 68
  VBFM 68
  VTCB 68
  XCNQ 68
  local storage option with ISWITCH 65
  LOGL latch 66
  long-term clear interval 24
  long-term clear interval (CLRL) 34
  LPA Page 57
  LPA Page (GFA) 59
  LQB pool latch 66
  LSO=Y 65
  LU 6.2 manager latch 66

M
maintenance assessment service
  MAS 78
MAS
  maintenance assessment service 78
MAXG 37
MAXG major 72
MDEX 38, 41, 46, 48, 55
  command output 46
  label field arguments 71
  MDEX display 44
MDEX/PDEX
  command format 44
  output examples 46
menu
  bottleneck analysis 46
  menu interface 18
  menus 18
model 3270 screen 40
MSM
  multi-services manager 78
MTHR 40, 47, 48
MTHR and SCAL commands output 41
MTHR minor 52, 70
multi-services manager MSM 78
MVS
Waits execution states group 56

O
OBFM latch 66
Open/Close latch 66
Other
   abend 66
   DL/I WAIT 67
   latch 67
   wait 67
overflow allowed 63

P
page faults 56, 58, 59, 60
pageable link pack area (PLPA) 57, 59
PDEX 38, 46, 48, 55
   command output 47
   display 44
   label field arguments 71
PI
   resource 67
   wait 67
printing problems 10
processing state 24
PSB group
   number supported 24
pseudo abend 67

Q
Queue Manager latch 67

R
READY queue 58
RESM 38
RESM minor 70
resuming collector sampling (RESM) 38

S
sample rate 24
   changing 39–41
sampling interval control (STIM) 39
SCAL 41, 47, 48
SCAL minor 70
scaling factor
   specifying 41
scaling factor for MDEX command (SCAL) 41
secondary index PCB 55
severity levels
   customer support 75
short-term clear interval 24
short-term clear interval (CLRS) 35
SMB Queue latch 67
SMGT latch 67
START Bottleneck Analysis 27
START DEXAN command 26
STIM 39
STIM minor 70
STOP ID=DX 32
SUSP 38
SUSP minor 70
suspending the collector (SUSP) 38
swapped in MPP 56
swapping in 56
sync point processing 64

T
TCT block latch 67
telephone support
   customer support 74
The 56
threads
   executing 19
THRS 41
   minor 52
THRS minor 48, 70
   command output 41
TM subqueue latch 68
transaction
   counts 44
   time 44
transaction group commands 72

U
Using CPU in
   APPL 53
   IMS 54

V
VBFM latch 68
VSAM buffer wait 68
VTCB pool latch 68

W
wait HSSP PVT pool 68
waits
   CPU 56
   database 55
MVS  52, 56

X

XCNQ latch  68
XCRB  64
X-MEM Page (GFA)  60