Note
Before using this information and the product it supports, be sure to read the information in "Notices" on page 247.

This edition applies to the following releases and to all subsequent releases and modifications until otherwise indicated in new editions:
• IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS, version 4, release 2, modification 0 (5655-Q07)
• IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS, version 4, release 2, modification 0 (5655-Q08)

This edition replaces SC18-9983-02.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
About this publication

This publication describes how to create reports and how to use these reports to assess the performance of a DB2® system. It also suggests methods to optimize your DB2 system.

After reading this publication, you should be able to select the report sets that are most appropriate for your requirements. You should also be familiar with the methods to create reports. The tuning information helps you interpret the information from the reports.

This publication is closely related to the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS; IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS: Report Command Reference, which describes the commands to create reports, and the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS; IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS: Report Reference, which shows and explains the reports.

Always check the IBM® DB2 and IMS™ Tools Library Web page and the Tivoli® library page for the most current version of this publication:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html
http://publib.boulder.ibm.com/tividd/td/tdprodlist.html

The technical changes for this edition are summarized under “Summary of changes” on page xv. Specific changes since the previous edition of this publication are indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

The product often provides context-related online help information that can be invoked from menus, panels, and windows by using the PF key F1 or the Help button. Online help information is not necessarily repeated in this publication, especially if it is very detailed information that is of interest only when you actively work with a function. You are encouraged to use F1 or Help to see the entire available information.

Who should read this publication

This publication is for IBM data server professionals who needs to produce and interpret OMEGAMON® XE for DB2 PE reports.

Conventions used in this information

This information uses several conventions for special terms and actions, and operating system-dependent commands and paths.

Panels and figures

The panels and figures in this document are representations. Actual product panels might differ.
Symbols

The following symbols might appear in command syntax:

Table 1. 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. You can use the argument on the left or the argument on the right. Example: YES</td>
</tr>
<tr>
<td>( )</td>
<td>Denotes optional arguments. Arguments that are not enclosed in square brackets are required. Example: APPLDEST DEST (ALTDEST) In this example, DEST is a required argument and ALTDEST is optional.</td>
</tr>
<tr>
<td>{}</td>
<td>Some documents use braces to denote mandatory 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>

Notation conventions

The following conventions are used when referring to high-level qualifiers:

$hilev$ A high-level qualifier. The high-level qualifier is the first prefix or set of prefixes in the data set name. Site-specific high-level qualifiers are shown in italics.

For example:
- $thilev$ refers to the high-level qualifier for your target data set
- $rhilev$ refers to the high-level qualifier for your runtime data set
  For members in target libraries, the high-level qualifier is $thilev$ rather than $rhilev$.
- $shilev$ refers to the SMP/E library high-level qualifier

Typeface conventions

This information uses the following typeface conventions:

**Bold**
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolon lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Note:**)
- Keywords and parameters in text

*Italic*
- Words defined in text
• Emphasis of words (for example: Use the word that to introduce a restrictive clause.)
• New terms in text (except in a definition list)

Monospaced
• Examples and code examples
• File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
• Message text and prompts addressed to the user
• Text that the user must type
• Values for arguments or command options

Significant elements
• **Recommendation**: provides guidance when more than one option is available.
• **Related reading**: refers you to other publications that contain relevant information.
• **Requirement**: identifies a condition that must be met to ensure that the product is functional.
• **Restriction**: identifies a restriction or limitation with this product or an associated procedure.

---

**How to read syntax diagrams**

The following rules apply to the syntax diagrams used in this information:

**Arrow symbols**
Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

- Indicates the beginning of a statement.
- Indicates that the statement syntax is continued on the next line.
- Indicates that a statement is continued from the previous line.
- Indicates the end of a statement.

**Conventions**
• SQL commands appear in uppercase.
• Variables appear in italics (for example, *column-name*). They represent user-defined parameters or suboptions.
• When entering commands, separate parameters and keywords by at least one blank if there is no intervening punctuation.
• Enter punctuation marks (slashes, commas, periods, parentheses, quotation marks, equal signs) and numbers exactly as given.
• Footnotes are shown by a number in parentheses, for example, (1).
• A _ symbol indicates one blank position.

**Required items**
Required items appear on the horizontal line (the main path).

- Indicates a required item.

**Optional items**
Optional items appear below the main path.
If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.

Multiple required or optional items
If you can choose from two or more items, they appear vertically in a stack. If you must choose one of the items, one item of the stack appears on the stack main path.

If choosing one of the items is optional, the entire stack appears below the main path.

Repeatable items
An arrow returning to the left above the main line indicates that an item can be repeated.

If the repeat arrow contains a comma, you must separate repeated items with a comma.

If the repeat arrow contains a number in parenthesis, the number represents the maximum number of times that the item can be repeated.

A repeat arrow above a stack indicates that you can specify more than one of the choices in the stack.

Default keywords
IBM-supplied default keywords appear above the main path, and the
Where to find information

The documentation for this product is provided in three formats: HTML, PDF, and BookManager®. You can access the documentation in several ways.

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center Web site. Access the Tivoli software information center by first going to the Tivoli software library at the following Web site:

http://publib.boulder.ibm.com/tividd/td/tdprodlist.html

From the Sort Order menu, select Product Category. From the alphabetical list, click O to access all of the IBM Tivoli OMEGAMON product manuals.

Note: If you print PDF documents on other than letter-sized paper, set the option in the File ➤ Print window that allows Adobe® Reader to print letter-sized pages on your local paper.

The IBM Software Support Web site provides the latest information about known product limitations and workarounds in the form of Technotes for your product. You can view this information at the following Web site:

www.ibm.com/software/support

Ordering publications

You can order many IBM publications, such as product manuals and IBM Redbooks®, online at the following Web site:


You can also order by telephone by calling one of these numbers:

• In the United States: 800-879-2755
• In Canada: 800-426-4968

In other countries, contact your software account representative to order Tivoli publications.

Accessing terminology online

The IBM terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the terminology Web site at the following Web address:

http://www.ibm.com/ibm/terminology/
Service updates and support information

You can access support information for this product in two ways:

- IBM DB2 Tools support Web page
- IBM Support Assistant

IBM DB2 Tools support Web page

Access the IBM DB2 Tools support Web page to find service updates and support information, including software fix packs, PTFs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads.

[www.ibm.com/software/data/db2imstools/support.html](http://www.ibm.com/software/data/db2imstools/support.html)

IBM Support Assistant

The IBM Support Assistant (ISA) is a free tool that provides access to several IBM support resources in a single location. Use the ISA tool to quickly access support-related information and serviceability tools for problem determination.

ISA offers the following components to help you with questions concerning the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS and IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS:

- Obtain quick access to appropriate IBM resources such as product pages, support pages and publication pages
- A search component, which helps you access pertinent Support information in multiple locations.

To begin using ISA, complete the following steps:

1. Download ISA from the following IBM Web page:
2. Start the ISA tool.

   **Note:** ISA runs as a Web application in the default system-configured Web browser.
3. Select the Updater tab.
4. Select the New Products and Tools tab.
5. The plug-in features are categorized by product family. Select Tivoli then IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS and IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS.
6. Check the feature(s) to be installed, then click on the “Install” button.
7. Restart ISA.

To learn more about how to use ISA, click the Help link in the IBM Support Assistant window.

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully.

The major accessibility features in this product enable users to do the following:
• Use assistive technologies, such as screen-reader software and digital speech synthesizer, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using those technologies with this product.
• Operate specific or equivalent features using only the keyboard.
• Magnify what is displayed on the screen.

In addition, the product documentation includes the following features to aid accessibility:
• All documentation is available in both HTML and convertible PDF formats to give the maximum opportunity for users to apply screen-reader software.
• All images in the documentation are provided with alternative text so that users with vision impairments can understand the contents of the images.

Navigating the interface by using the keyboard

Standard shortcut and accelerator keys are used by the product and are documented by the operating system. Refer to the documentation provided by your operating system for more information.

Magnifying what is displayed on the screen

You can enlarge information in the product windows using facilities provided by the operating systems on which the product is run. For example, in a Windows® environment, you can lower the resolution of the screen to enlarge the font sizes of the text on the screen. Refer to the documentation provided by your operating system for more information.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this information or any other documentation:
• Use the online reader comment form located at:
  There you will find the feedback page where you can enter comments and send them.
• Send your comments by e-mail to swsidd@de.ibm.com. Be sure to include the documentation name, the part number, the version number, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
• Print and fill out the reader comment form located at the back of this information. You can either give the completed form to your local IBM branch office or IBM representative, or you can send it to the address printed on the reader comment form.
Summary of changes

This topic summarizes the significant improvements or enhancements for the product and refers you to the relevant topics for more information.

SC19-2510-00 — April 2009

- A topic about troubleshooting empty reports was added.
- The following technical terms were changed: “OMEGAMON Server” is now “OMEGAMON Collector”, “PE Server” is now “PE Server subtask”.
- A comprehensive description of the usage aspects of the CALCULATE and OPTIMIZE subcommand options for the ACCOUNTING REDUCE command was added. These options can be used to calculate required ACWORK space and to control the performance of REDUCE processing.

SC18-9983-02 — June 2008

- The Near-Term History Data Collector was added to the list of methods you can use to collect data for OMEGAMON XE for DB2 PE reports and traces. It can be configured to store collected data in sequential data sets, Generation Data Group (GDG) data sets, and VSAM data sets.
- Extensions to the ACCOUNTING FILE DATATYPE command with regard to loading Accounting data from several data sets into a DB2 table are reflected.
- The topic about tailoring report layouts was revised.
Chapter 1. Overview

The following information provides an overview of the OMEGAMON XE for DB2 PE product and the OMEGAMON XE for DB2 PE batch reporting.

Introduction to OMEGAMON XE for DB2 PE

This following information introduces the main functions and components of OMEGAMON XE for DB2 PE.

The following topics provide additional information:
- “What does OMEGAMON XE for DB2 PE do?”
- “Components of OMEGAMON XE for DB2 PE” on page 2
- “Overview of Performance Expert Agent for DB2 Connect Monitoring” on page 3

What does OMEGAMON XE for DB2 PE do?

OMEGAMON XE for DB2 PE integrates performance monitoring, reporting, buffer pool analysis, and a Performance Warehouse function into a single tool. It provides a single overview system that monitors all subsystems and instances for different operating systems in a consistent way.

OMEGAMON XE for DB2 PE supports DB2 on z/OS® and in the following multiplatform environments:
- Windows
- AIX
- HP-UX
- Linux®
- Linux/390
- Solaris Operating Environment

The following list summarizes the advanced capabilities of OMEGAMON XE for DB2 PE. The availability of these capabilities, however, varies, depending on whether you install Performance Expert for Multiplatforms, Performance Expert for Workgroups, OMEGAMON XE for DB2 PE for z/OS, or one of the stand-alone products OMEGAMON XE for DB2 PM or DB2 Buffer Pool Analyzer. For a detailed description of the different capabilities, see the following Web site:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

In general, OMEGAMON XE for DB2 PE includes the following advanced capabilities:
- Provides detailed analysis of key performance factors to control and tune the performance of DB2 and DB2 applications.
- Provides a real-time online monitor, a wide range of reports, expert analysis, and an Explain feature to analyze and optimize SQL statements.
- Provides a simulation function for certain buffer pool tuning actions before you change your system.
- Provides a Performance Warehouse function for storing performance data and analysis functions.
- Enables you to monitor Database Connection Services (DCS) connections by means of Performance Expert Agent for DB2® Connect™ Monitoring (PE Agent).
Components of OMEGAMON XE for DB2 PE

OMEGAMON XE for DB2 PE consists of several interacting components. The components vary dependent on the platform in use.

The main components of OMEGAMON XE for DB2 PE are:

- Performance Expert Client
- OMEGAMON Collector for z/OS
- Performance Expert Server for Multiplatforms
- Performance Expert Agent for DB2 Connect Monitoring (PE Agent)

The following figure shows how these components work together:

Performance Expert Client represents the user interface to Performance Expert for Multiplatforms, Performance Expert for Workgroups, and OMEGAMON Collector for z/OS.

The following list shows what the client comprises when you install the different programs:

- If you install Performance Expert for Multiplatforms, the client comprises performance monitoring, reporting, buffer pool analysis, and Performance Warehouse.

**Note:** The installation procedure for Performance Expert for Multiplatforms and Performance Expert for Workgroups is identical.
If you install OMEGAMON XE for DB2 PE for z/OS, the client comprises OMEGAMON XE for DB2 PM, Buffer Pool Analyzer, and Performance Warehouse.

If you install OMEGAMON XE for DB2 PM, the client comprises Workstation Online Monitor and Performance Warehouse.

If you install Buffer Pool Analyzer, the client comprises only Buffer Pool Analyzer.

To get information about performance data, the server components access the corresponding database engines.

- OMEGAMON Collector for z/OS accesses DB2 for z/OS and OS/390®.
- Performance Expert Server for Multiplatforms accesses DB2 for AIX, HP-UX, Linux, the Solaris Operating Environment, and Windows.

The following section provides an overview of Performance Expert Agent for DB2 Connect Monitoring (PE Agent).

**Overview of Performance Expert Agent for DB2 Connect Monitoring**

Performance Expert Agent for DB2 Connect Monitoring (PE Agent) monitors Database Connection Services (DCS) connections within the Distributed Relational Database Architecture™ (DRDA®) of DB2.

PE Agent is required if you want to monitor access of a remote Performance Expert Client to mainframe DB2 databases.

If PE Agent is installed on the system on which DCS connections are performed, it collects connection-related data, such as the status of a DCS connection and statistics about DB2 Connect activities. The collected data is stored in the DB2PM database on an OMEGAMON Collector.

You can use the collected data to find out whether the DCS connection is working or to identify network problems between DB2 Connect and the DB2 data server on z/OS.

The following figure provides an overview of how PE Agent is integrated in a system environment. It shows the relationship of one PE Agent to one OMEGAMON Collector.
You can set up PE Agent and OMEGAMON Collector in one of the following ways:

- One PE Agent can send data to one or more OMEGAMON Collectors.
- One OMEGAMON Collector can receive data from one or more PE Agents.

### Introduction to OMEGAMON XE for DB2 PE batch reporting

The batch reporting facility presents historical information about the performance of the DB2 system and applications in reports and data sets. System performance data shows information about topics such as CPU times, buffer pool usage, locking, log activity and I/O activity. Application data shows how individual programs behave in DB2.

You can examine a DB2 system and its applications in real time by using the workstation client. You can also use the workstation client to launch and control OMEGAMON XE for DB2 PE reporting functions.

The Performance Warehouse is an environment where you can define, schedule, and run processes that automate the creation of reports. You can also use processes to convert and load these reports or other OMEGAMON XE for DB2 PE data sets into a Performance Database. In the Performance Warehouse you can analyze the data in a Performance Database by using rules.

The following topics provide additional information:

- "What OMEGAMON XE for DB2 PE does" on page 8
- "Input for OMEGAMON XE for DB2 PE reports" on page 8
- "The OMEGAMON XE for DB2 PE reporting process – an overview" on page 8

### What OMEGAMON XE for DB2 PE does

OMEGAMON XE for DB2 PE uses DB2 instrumentation data to generate performance reports in a form that is easy to understand and analyze.
You can use OMEGAMON XE for DB2 PE to:
- Determine DB2 subsystem performance and efficiency
- Identify and resolve potential problems
- Tune the DB2 subsystem
- Measure an application's performance and resource cost
- Tune applications and SQL queries
- Assess an application’s affect on other applications and the system
- Gather information for cost purposes

OMEGAMON XE for DB2 PE provides information at various levels of detail depending on your needs.

**Output types**
OMEGAMON XE for DB2 PE generates reports, traces, data sets, and logs.

**Reports and traces:**
Reports show summarized DB2 events and traces show individual DB2 events.

*Reports* show DB2 events summarized by OMEGAMON XE for DB2 PE identifiers, such as primary authorization ID or plan name. For example, you can produce an Accounting report that shows all threads summarized for every individual plan.

*Traces* show individual DB2 events, for example, for a particular thread. Depending on the report set that you request, these could include thread terminations, grants of privileges, deadlocks, or utility executions. All events are listed individually, usually in the order of occurrence.

**Data sets:**
Formatted data can be stored in data sets that are suitable for loading into DB2 tables.

The data in DB2 tables can be used in different ways, for example, it can be used to produce tailored reports using a reporting facility such as the IBM Query Management Facility (QMF™). You can also load reports and traces into OMEGAMON XE for DB2 PE’s Performance Database for additional analysis.

**Logs:**
OMEGAMON XE for DB2 PE can log several activities and keep this information available in defined data sets.
- *The Exception Log* contains Accounting and Statistics records with at least one field outside user-specified thresholds.
- *The IFCID Frequency Distribution Log* provides counts of input records by IFCID.
- *The Job Summary Log* provides a summary of events during OMEGAMON XE for DB2 PE execution.
- *The DPMLOG Execution Log* contains OMEGAMON XE for DB2 PE processing messages.

**Report sets**
OMEGAMON XE for DB2 PE generates various reports and traces that provide performance information about different areas of interest (also called *report sets*) and of various levels of detail.
**Accounting reports and traces:**

Accounting reports and traces summarize information about DB2 resource activity associated with particular DB2 applications.

Use this report set to:
- Obtain summarized information about DB2 local and distributed activity associated with the execution of DB2 plans.
- Identify potential performance problem areas in local and distributed activity.
- Track trends in DB2 resource usage.

Accounting reports show the efficiency of the subsystem or application and often provide enough information for you to resolve performance problems.

**Statistics reports and traces:**

Statistics reports and traces summarize information that is collected by the DB2 Instrumentation Facility for an entire DB2 subsystem. The collected data is logged at intervals that are specified when you install DB2.

Use this report set to:
- View system-wide statistics for key DB2 components.
- Compare system performance in several reporting intervals.
- Assess system-wide performance that can be summarized in a single report.

Statistics reports show the efficiency of the subsystem or application and often provide enough information for you to resolve performance problems.

**SQL Activity reports and traces:**

SQL Activity reports and traces show information about SQL activities that occur during the processing of a DB2 application.

Use this report set to:
- Analyze SQL calls within a logical unit of work.
- Analyze the access paths selected by the DB2 optimizer for local SQL calls.
- Observe the Data Manager scans, locking, buffer manager I/O and other related DB2 activity that occur during the execution of SQL requests.
- Collect the SQL activity according to author, plan name, or other OMEGAMON XE for DB2 PE identifiers.

**Locking reports and traces:**

Locking reports and traces show detailed information about locking activities within DB2.

Use this report set to:
- Obtain summarized information about user activities related to lock suspensions and lockouts.
- Obtain information about every lock request made by DB2.
- Obtain information about every suspension, lockout, deadlock, or timeout during a specified interval.
I/O Activity reports:

I/O Activity reports show information about I/O activities performed throughout the DB2 subsystem.

Use this report set to:
- Track I/O volumes and service times.
- Obtain summarized information about the active log, archive log, bootstrap data set (BSDS), buffer pool, and EDM pool.

Audit reports and traces:

Audit reports and traces show information about access to DB2 resources.

Use this report set to:
- Obtain information about who performed an audited action, when the action was performed, and what auditable DB2 objects were involved.
- Track DB2 resource access by OMEGAMON XE for DB2 PE identifier, and the granting and revoking of DB2 privileges.
- Identify security breaches and violations.

Utility Activity reports and traces:

Utility Activity reports and traces show information about utility and bind activities during the processing of a DB2 application.

Use this report set to:
- Analyze bind and utility activity within a logical unit of work.
- Associate DB2 work with individual users or OMEGAMON XE for DB2 PE identifiers.
- Recognize long utility or bind times, which can help you to identify bottlenecks in the system.

Record Trace reports:

Record Trace reports show the contents of selected instrumentation records.

Use this report set to format a subset of data obtained from Statistics, Accounting, or Performance trace information.

Explain reports:

Explain reports show information about the access path that is selected by DB2 for a particular SQL statement.

Use this report set to determine access path problems or problems with an application’s design.

System Parameters reports:

System Parameters reports show information about the configuration of your DB2 subsystem.
The report shows values for all DB2 system parameters that were in effect at the time the performance data was collected.

Use this report set to determine system parameters that you might want to change during tuning activities.

**Levels of detail of report sets:**

Report sets provide different levels of details, ranging from high-level summaries to individual DB2 events.

The level of detail and the amount of data varies by report set and output type, as shown in Figure 3. Generally, start any problem determination task with high-level summary reports, such as Accounting and Statistics reports. Use traces if you require information about individual DB2 events.

**Input for OMEGAMON XE for DB2 PE reports**

The DB2 trace facility, also called the DB2 instrumentation facility, gathers information about the system. After DB2 has collected and externalized this data, you can use it to generate reports and data sets.

An easy way to produce a DB2 trace command is to use the Traces function of the Performance Expert workstation client. You can use this facility to create and manage DB2 traces. You can specify what types of reports you want to produce and how and when the traces should start and stop.

**The OMEGAMON XE for DB2 PE reporting process – an overview**

The OMEGAMON XE for DB2 PE reporting process uses various input sources to produce the different output types.

Figure 4 on page 9 summarizes the main elements and functions involved in OMEGAMON XE for DB2 PE processing. The top part of the figure shows the various inputs to OMEGAMON XE for DB2 PE and the bottom part of the figure shows the different output types.

Solid lines indicate input or output, and broken lines indicate specifications the user makes. Words written in uppercase next to the arrows are subcommands.
For information about input data sets, read "Collecting data for reports" on page 11, and for information about DPMPARMS, read "Customizing OMEGAMON XE for DB2 PE functions" on page 198. Exception thresholds are described in "Using Statistics and Accounting reports to identify exceptions" on page 58.
Chapter 2. Generating OMEGAMON XE for DB2 PE reports

The following information describes how to collect performance data and how to create reports from this data.

**Note:** OMEGAMON XE for DB2 PE provides sophisticated and flexible methods to collect performance data and to create reports from collected data. For both the collection of data and the creation of reports, you have several choices to determine which data to collect and which data to include in reports. The basic rule is that your final reports will only show data that has previously been collected and has not been filtered out by any of the command parameters. Empty reports, processing counts of zero, or messages showing zero processed records usually indicate that required data within specified or required time intervals is not available in your input data or has been filtered by inappropriate command parameters during report generation. If required, refer to “Troubleshooting empty reports” on page 240 for more information.

Collecting data for reports

Before you can produce an OMEGAMON XE for DB2 PE report, you need to collect data from DB2 as input. The DB2 Instrumentation Facility Component (IFC) provides a trace facility that is used to record DB2 data and events.

Use any of the following methods to collect data for OMEGAMON XE for DB2 PE reports and traces:

- The Workstation Online Monitor to manage DB2 traces.
- The ISPF Online Monitor to configure a Collect Report Data (CRD) task that collects report data.
- The Data Warehouse to create a process to collect report data.
- DB2 installation system parameters to start traces at DB2 startup time.
- DB2 -START TRACE commands from the console.
- The Collect Report Data Batch program, which is a flexible and resource-efficient way to collect and postprocess report data.
- The Near-Term History Data Collector, which can store collected data in sequential data sets.

The following topics provide additional information:

- "Managing DB2 traces with the Workstation Online Monitor"
- "Collecting report data with the ISPF Online Monitor" on page 12
- "Collecting report data by specifying DB2 startup parameters" on page 21
- "Collecting report data with the DB2 START TRACE command" on page 21
- "Collecting report data with the Collect Report Data Batch program" on page 28
- "Collecting report data with the Near-Term History Data Collector" on page 30

Managing DB2 traces with the Workstation Online Monitor

The easiest way to produce input data for reports is to use the Workstation Online Monitor. You need to specify the report set for which you want to collect data and the data set where it is to be collected. Then, you can use this data set as input for reports.
You can start the collection of report data manually, or you can set it up to start automatically based on certain conditions, such as:
- A specified point in time
- When an exception threshold is reached
- An exception event is encountered

You can configure and start collect tasks for:
- One or more DB2 PM report sets
- Specific types of reports or traces within a report set
- Specific IFCIDs

You can also limit the data to specific:
- Locations
- Plan names
- Authorization IDs

You can configure DB2 traces to start:
- At a specified time of the day
- When a specified periodic exception is detected
- When a specified exception event occurs
- Immediately

You can stop DB2 traces manually or configure traces to stop:
- After a specified elapsed time
- After a specified number of trace records have been collected
- After a thread has been terminated or reused
- After a particular IFCID has been collected a specified number of times

For details about running DB2 traces, see Monitoring Performance from Performance Expert Client.

**Collecting report data with the ISPF Online Monitor**

You can use the ISPF Online Monitor to write performance data to a data set that is to be used as input for reports. This can be useful when you want to avoid flooding SMF with large numbers of DB2 trace records, and make DB2 trace data immediately available without affecting SMF or GTF record collection.

To collect data, you require the necessary DB2 authority to start and stop DB2 traces.

To collect performance data, you first configure a collect task. In each collect task, you specify the type of data that you want to gather, the trace start and stop criteria, and the output data set name. When you have configured and started a task, it triggers the appropriate DB2 traces to start and stop when the trace start and stop criteria have been met, and writes the collected data to the output data set.

You can configure and start up to four independent collect tasks. With each task, you can collect trace data for:
- One or more reports sets
- Specific types of reports or traces within a report set
- Specific IFCIDs

In addition, you can limit the data to specific:
- Locations
- Plan names
Authorization IDs

You can configure DB2 traces to start:
- At a specified time of the day
- When a specified periodic exception is detected
- When a specified exception event occurs
- Immediately

You can stop DB2 traces manually or configure traces to stop:
- After a specified elapsed time
- After a specified number of trace records have been collected
- After a thread has been terminated or reused
- After a particular IFCID has been collected a specified number of times

Accessing the Collect Report Data (CRD) panels

You can use the ISPF Online Monitor and its CRD panels to configure a collect task and to start and stop a collect task.

About this task

Before you can access the CRD panels to configure a collect task, the ISPF Online Monitor must be invoked from the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS main menu. Choose option 3 (View online DB2 activity - PE ISPF OLM) to invoke the ISPF Online Monitor, if not already active.

From the Online Monitor Main Menu, select option 6 (Collect Report Data) or 6a (Collect Report Data - General). Alternatively, enter COLLECT on the command line, or press F17 (Collect).

You can access the collect report data panels from within the Online Monitor by selecting option 6 (Collect Report Data) from the Online Monitor Main Menu.

The Collect Report Data panel is displayed:

```
DGOMAP00 Collect Report Data
PM01DLOC DSN1 V7

For any trace task enter one of the following actions:
1=Configure
2=Start
3=Display
4=Stop

Task Description                   Status
Collect data for acct/stats/audit___ Not yet started
Test case for buffer overruns_____ Collecting data
Collect Task C___________________ Never configured
Collect Task D___________________ Never configured

Command ===>
F1=Help    F2=Split    F3=Exit    F9=Swap    F12=Cancel    F16=Look
```

Figure 5. Collect Report Data panel

From the Collect Report Data panel you can configure and control the collection of report data.
• Option 1 (Configure) displays a window where you configure collect tasks to collect report data and where you can limit the collection of DB2 trace data to time periods or events of interest. Refer to “Configuring a collect task” for more details.

• Option 2 (Start) displays a window where you start a collect task after it has been configured. Starting a collect task actually starts a DB2 trace and enables the start and stop criteria that were specified with option 1 (Configure). Refer to “Starting and stopping traces” on page 18 for more details.

• Option 3 (Display) displays a window that shows the status of a collect task and any messages issued by that collect task. Refer to “Displaying trace status and messages” on page 20 for more details.

• Option 4 (Stop) displays a window where you can stop a collect task and associated DB2 traces. Stopping a collect task prevents the triggering of DB2 traces and stops all traces that were previously started by the task. Refer to “Starting and stopping traces” on page 18 for more details.

The Task Description column shows the description of each collect task. You can change a description by overtyping it with another description. The Status field shows the current status of the task.

If you see a task error in the Status field, use the DB2 DISPLAY TRACE command to check if the traces are still running. If necessary, use the DB2 STOP TRACE command to stop the trace. Before you restart the appropriate Collect Report Data function, exit the Collect Report Data panel (by pressing F3) and, in a second step, exit the Online Monitor Main Menu (by pressing F3 again) to return to the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS main menu. This way you stop the four asynchronous tasks that were automatically set up when you first selected option 3 (View online DB2 activity - PE ISPF OLM) from the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS main menu.

**Configuring a collect task**

Use the following windows to specify the data types and IFCIDs to be collected.

**About this task**

To configure a collect task, type 1 (Configure) next to a collect task in the Collect Report Data panel and press Enter. The Trace Configuration window is displayed:
Use this window to specify the trace trigger method, the report sets, and the types of data to be collected.

Before you can start a collect task, specify how the trace is to be triggered. You can specify that the DB2 traces are triggered by time (1), periodic exception (2), exception event (3), or started immediately (4).

You also need to select the report sets for which you want to collect data. When the trace start criteria have been met, the appropriate DB2 traces are started to collect data required for these report sets. A greater than symbol (>) in the selection field indicates report sets that were previously selected.

Use the fields on the bottom part of this panel to specify whether to restrict the collection of data to specific data types (see "Restricting data types for selected report sets"), IFCIDs (see "Restricting IFCIDs for selected report sets" on page 16), or OMEGAMON XE for DB2 PE identifiers (see "Restricting identifiers for selected report sets" on page 17). If any of these fields are selected, the appropriate windows are displayed where you can fill in the data collection criteria.

Use the OP Buffer size field to allocate the number of KB to the OPn buffer that is used for collecting the data. The valid range is 8 - 1024.

**Restricting data types for selected report sets:**

Use the Data to Collect window to select the data types to be collected by the collect task for a specific report set.

About this task

If you selected the Data Type field on the Trace Configuration window, the Data to Collect window [Figure 7] is displayed once for each selected report set that has more than one data type.

The Data to Collect window shown in [Figure 7] uses an example for the Audit report set.

![DGOMAP31 Data to Collect Row 1 to 7 of 7](image)

Task description . . . . . : Collect data for acct/stats/audit

Report
Set . . . : Audit

Enter one or more selection characters to start DB2 traces for specific data types or overtype with a blank to delete the selection.

- Select/Deselect all
/ Audited DDL Access
- Audited DML Access
> Audited DML at Bind Access
- Audited Utility Access
/ Authorization Change
- Authorization Control
- Authorization Failures
-- End of Data Types --

Command ==> __________________________________________________________
F1=Help F2=Split F3=Exit F7=Up F8=Down F9=Swap
F12=Cancel F16=Look

Figure 7. Data to Collect Window

If you select the Select/Deselect all field, all data type fields in this window are selected. If you type a blank in the Select/Deselect all field, all selections are deleted.

Select the data types and press Enter to process the changes. Press Enter again to proceed to the next panel.

Restricting IFCIDs for selected report sets:

Use the IFCID Selection window to exclude certain IFCIDs that would normally be collected for the selected report sets and data type.

About this task

If you selected the IFCID field on the Trace Configuration window, the IFCID Selection window [Figure 8 on page 17] is displayed.

Note: The IFCID Selection window is not shown if only one IFCID was collected for the previous selections.
You can use the Select/Deselect all field to select or deselect all fields in this panel.

Select the IFCIDs and press Enter to process the changes. Press Enter again to proceed to the next panel.

Restricting identifiers for selected report sets:

Use the Trace Qualification window to filter the data to be collected by the collect task.

About this task

If you selected the Requesting Location, Plan name and Authid field in the Trace Configuration window, the Trace Qualification window (Figure 9 on page 18) is displayed. Only data from threads that match the trace qualification criteria in this window will be collected. These trace qualification criteria are also used if thread termination is indicated in the Trigger by Time window (Figure 10 on page 19).
Specify the name of the requesting location, plan name, and authorization ID, then press Enter. Do not specify multiple entries for more than one identifier, otherwise the number of DB2 traces started by the collect task could exceed the DB2 limit of 32 traces.

Trace qualification does not apply to all IFCIDs. Some system-related traces are collected regardless of the trace qualification criteria specified in the Trace Qualification window, for example, traces for IFCIDs 1, 2, 4, 104, 105, 106, and 202.

Starting and stopping traces
You can trigger DB2 traces to start by time, periodic exception, exception event, or immediately. After you specified the trace criteria, a window is displayed where you complete the start criteria and define the stop criteria.

About this task
Which window is displayed depends on what you specified in the Trigger by field on the Trace Configuration window. Use these windows to specify the criteria that must be met before the collect task is automatically started or stopped. These windows are the same in appearance except for the start trigger specification section.

Because of the possibility of output buffer overruns, you need to specify one of the stop conditions: Elapsed time or Number of records collected. Records can get lost if a buffer overrun occurs.

The Trigger by Time window [Figure 10 on page 19] is displayed if you specified that the trace is triggered by time. Use this window to specify a particular output data set name, and to set the start and stop trigger criteria for the collect task. You can set the DB2 traces to start at a specified time, and to stop after a specified number of minutes have passed, a specified number of records or IFCIDs have been collected, or a thread matching the trace qualification criteria has terminated.
All trace data collected by the collect task is written to the data set specified in this window. If you specify a disposition of 3 (New), the data set is dynamically allocated with the following attributes:

- **RECFM:** VBS
- **LRECL:** 32756
- **BLKSIZE:** 6233

If you want to create the data set manually, it should have a variable record format and a record length (LRECL) of at least 4092.

Specify the time you want the DB2 traces to start for this task.

Select one or more of the trace stop triggers shown in this window and enter the required criteria for those triggers. The trace is stopped when any stop criteria is satisfied.

After you specified the required criteria in this window, press Enter to process the new values. Press Enter again or F3 (Exit) to return to the Collect Report Data panel.

The other trigger windows are the same as this window except for the start trigger section.

- In the Trigger by Periodic Exception window, you can set the DB2 traces to start when a specified periodic exception has occurred.
- In the Trigger by Exception Event window, you can set the DB2 traces to start when a specified exception event has occurred.

**Figure 10. Trigger by Time window**

All trace data collected by the collect task is written to the data set specified in this window. If you specify a disposition of 3 (New), the data set is dynamically allocated with the following attributes:

**RECFM:**

- VBS

**LRECL:**

- 32756

**BLKSIZE:**

- 6233

If you want to create the data set manually, it should have a variable record format and a record length (LRECL) of at least 4092.

Specify the time you want the DB2 traces to start for this task.

Select one or more of the trace stop triggers shown in this window and enter the required criteria for those triggers. The trace is stopped when any stop criteria is satisfied.

After you specified the required criteria in this window, press Enter to process the new values. Press Enter again or F3 (Exit) to return to the Collect Report Data panel.

The other trigger windows are the same as this window except for the start trigger section.

- In the Trigger by Periodic Exception window, you can set the DB2 traces to start when a specified periodic exception has occurred.
- In the Trigger by Exception Event window, you can set the DB2 traces to start when a specified exception event has occurred.
• In the Trigger Immediately window, there are no start criteria because the DB2 traces are started immediately when the respective collect task is started in the Collect Report Data panel.

You can obtain a list of IFCIDs or exception field names by positioning the cursor under any field with a trailing plus sign (+) and pressing F4 (Prompt).

**Displaying trace status and messages**

Use the Trace Status Summary window to view the status of a collect task in detail and any messages issued by that task.

**About this task**

To view the status of a trace, type 3 (Display) next to the collect task in the Collect Report Data panel.

![Figure 11. Trace Status Summary window](image)

From this window, you can select the Display Status Detail field to display the Trace Status Detail window, where you can view further details about the status of the collect task.

You can also select the Display messages for this task field to display the Trace Messages window, where you can view the trace messages generated by the collect task. Messages for all collect tasks are kept for the duration of your Online Monitor session.

The Active Traces for this Destination section of this window lists all active DB2 traces started by the task, and shows the trace type, trace class, output buffer destination, and qualification criteria for each active DB2 trace started by the task. If many DB2 traces are listed, use the scrolling keys F7 (Up) and F8 (Down) to browse the list.
**DB2 trace termination**

You are notified if collect tasks are active when you exit the ISPF Online Monitor or when you change DB2 subsystems.

If any collect tasks are active when you exit the ISPF Online Monitor, you are notified by one of the Asynchronous Task Termination panels. You can either exit the ISPF Online Monitor and terminate all asynchronous tasks, or return to the Online Monitor Main Menu keeping all asynchronous tasks active.

If any collect tasks are active when you change DB2 subsystems, you are notified by one of the Asynchronous Task Termination panels. You can either change DB2 subsystems and terminate all asynchronous tasks, or return to the previous panel keeping all asynchronous tasks active.

**Collecting report data by specifying DB2 startup parameters**

For regular monitoring, you can set the trace facility to automatically start Accounting, Statistics, and Audit traces when DB2 is started by using the DB2 Tracing panel (DSNTIPN).

You can modify the parameters in the DB2 Tracing panel (DSNTIPN) to indicate which types of data you want to trace. You can specify these values when you install, migrate, or update DB2.

**Collecting report data with the DB2 START TRACE command**

You can use the DB2 START TRACE command to obtain all types of DB2 trace data.

You need TRACE privilege or SYSOPR, SYSCTRL, or SYSADM authority to issue the command.

You can enter the command from an OS/390 console, the DSN command processor, the DB2I commands panel, an IMS or CICS® terminal, or the OMEGAMON XE for DB2 PE Online Monitor.

The following diagram shows the command syntax.
The START TRACE command accepts the following parameters and options:

**TRACE**
- Specifies the type of data to be traced.
  - P  Performance data
  - A  Accounting data
  - S  Statistics data
  - AU Audit data
  - G  Global data

**DEST**
- Specifies the destination to which the traced data is directed. The destination can be the SMF, the GTF, or an OP buffer.

  SMF and GTF are service programs that provide a means of recording performance data. SMF is usually used for continuous monitoring and is the default destination for Statistics, Accounting, and Audit traces. GTF is usually used for monitoring a specific problem when the amount of data is large.

If you want to direct a large volume of data to SMF, check the SMF buffer sizes to see whether they need to be increased.

If you specify SMF as destination:
1. Ensure that SMF is active before you start the trace. For more information, see z/OS MVS System Management Facilities (SMF).
2. Ensure that SMF collects the following records:
   - DB2 Accounting records (SMF type 101 records)
   - DB2 Audit records (SMF type 102 records)
   - DB2 Statistics records (SMF type 100 records)
   - DB2 Performance records (SMF type 102 records)
3. Ensure that the SMF data sets and buffers are large enough to hold the data being collected. If the SMF buffers run out of space, SMF rejects any records while the shortage of space exists. Lost data cannot be recaptured. However, a long Statistics report or trace shows the number of records that have been lost in the DB2 IFC Destination block of the report.
**CLASS**
Specifies one or more trace classes within the trace type.

**IFCID**
Specifies one or more IFCIDs to be traced in addition to those contained in the specified trace classes.

If you want to exclusively trace the IFCIDs specified in the IFCID option, use trace classes 30-32. These trace classes have no predefined IFCIDs. Use this option with care, many reports have dependencies on several IFCID combinations. When the proper records are missing, OMEGAMON XE for DB2 PE cannot make the proper record relationships to produce a meaningful report.

**AUTHID and PLAN**
Specifies authorization IDs and plan names to be traced.

Consider how you want to use the traced data. If you want to examine a specific problem, choose the authorization IDs and plan names based upon the application and users of the application to be monitored. If you are doing daily monitoring, select all AUTHIDs and PLANs to get an overview of the activity on your system.

**TDATA**
Specifies the product section headers to be placed into the product section of each trace record. The product section of a trace record can contain multiple headers.

If you do not specify TDATA, the type of trace determines the type of product section header. By default, correlation headers and distributed headers (if present) are included. However, specifying CPU overrides the default so that only CPU headers are included. If you want CPU, correlation, and distributed headers, specify all.

All IFC records have a standard IFC header. The correlation header is added for Accounting, Performance, Audit, and Monitor records. The trace header is added for serviceability records. The headers relevant to OMEGAMON XE for DB2 PE are:

<table>
<thead>
<tr>
<th>Report set</th>
<th>COR</th>
<th>CPU</th>
<th>DIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting</td>
<td>Required</td>
<td>Not used</td>
<td>Required if distributed data is present</td>
</tr>
<tr>
<td>Audit</td>
<td>Required</td>
<td>Not used</td>
<td>Required if distributed data is present</td>
</tr>
<tr>
<td>I/O Activity</td>
<td>Required</td>
<td>Not used</td>
<td>Required if distributed data is present</td>
</tr>
<tr>
<td>Locking</td>
<td>Required</td>
<td>Not used</td>
<td>Required if distributed data is present</td>
</tr>
<tr>
<td>Record Trace</td>
<td>Used if present</td>
<td>Used if present</td>
<td>Required if distributed data is present</td>
</tr>
<tr>
<td>SQL Activity</td>
<td>Highly recommended</td>
<td>Recommended</td>
<td>Required if distributed data is present</td>
</tr>
</tbody>
</table>
Table 2. Product section headers relevant to OMEGAMON XE for DB2 PE (continued)

<table>
<thead>
<tr>
<th>Report set</th>
<th>COR</th>
<th>CPU</th>
<th>DIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics</td>
<td>Not used</td>
<td>Not used</td>
<td>Required if distributed data is present</td>
</tr>
<tr>
<td>Utility Activity</td>
<td>Required</td>
<td>Recommended</td>
<td>Required if distributed data is present</td>
</tr>
</tbody>
</table>

**LOCATION**

Specifies locations with distributed relationship to be traced. If specified, allied threads are excluded from the trace. Only allied-distributed threads and DBATs associated with the specified locations are traced. If you omit the LOCATION option, all threads (including allied threads) from all locations are traced.

You can specify up to eight locations. A separate trace is started for each location.

If you specify more than one PLANNAME or AUTHID, you can specify only one location.

You can specify the LUNAME of non-DB2 systems, in the form `<luname>`, in place of a location name.

LOCATION has no effect when Accounting class 2 is started.

**START TRACE command examples**

The command examples show how to specify the different options of the START TRACE command.

If you do not specify a trace class, a default trace class is used. So, if you specify the following, data is collected only for Statistics class 1.

```
-START TRACE(S)
```

To collect Audit class 2 data, enter:

```
-START TRACE(AU) CLASS(2)
```

To collect only specific IFCIDs within a trace type, specify one of trace classes 30, 31, or 32, which are installation defined and contain no predefined IFCIDs. The following example shows how to collect only IFCIDs 44 and 45 (lock suspensions):

```
-START TRACE(P) CLASS(30) IFCID(44,45)
```

To collect Performance class 16 information and IFCID 68 and 69 data, specify:

```
-START TRACE(P) CLASS(16) IFCID(68,69)
```

**Note:** The specified IFCIDs must belong to the trace type that you specified, otherwise no data is collected for these IFCIDs.

If you want to start all trace classes of Accounting data, you can either use an asterisk or specify all trace classes.

```
-START TRACE(A) CLASS(*)
-START TRACE(A) CLASS(1,2,3,5,7,8)
```
The default destination for Accounting, Statistics, and Audit trace types is SMF, but you can route the trace data to GTF by specifying DEST(GTF) or to both SMF and GTF by specifying DEST(SMF,GTF) in the START TRACE command.

If you omit the TDATA option, correlation headers and distributed headers (if present) are included by default. However, specifying CPU overrides the default so that only CPU headers are included. If you want CPU, correlation, and distributed headers, specify all as in the following example:

```
-START TRACE(P) CLASS(1,2,3) DEST(GTF) TDATA(CPU,COR,DIST)
```

**DB2 instrumentation data**

Understand how DB2 instrumentation data is grouped and how these groups relate to OMEGAMON XE for DB2 PE report sets. With this information you can correctly specify the data to be collected with the DB2 START TRACE command.

Each DB2 event is recorded by its Instrumentation Facility Component (IFC) as trace record. These trace records have unique IFC identifiers (IFCID). To have these IFCIDs externalized, appropriate trace types must be started.

Most IFCIDs are grouped into trace classes. A trace class defines a certain group of events or data within a trace type. Each class consists of one or more IFCIDs. You can limit the amount of data to be collected by specifying only certain classes for a type.

The instrumentation data types used as input to OMEGAMON XE for DB2 PE reporting facility are:

- **Statistics** data shows to what extend the DB2 system services and database services are used. You can use this information to plan DB2 capacity and to tune an entire set of DB2 programs. Statistics data also contains information about deadlocks, timeouts, and DDF exception events.
  
  The Statistics trace is written at specified intervals. You can control the Statistics interval by using STATISTICS TIME in the DSNTIPB installation panel.

- **Accounting** data provides information related to application programs and packages.
  
  A DB2 Accounting trace record starts when a thread is allocated to DB2 and ends when the thread terminates, is reused, or becomes inactive. You can use Accounting data for program-related tuning and to assess DB2 usage for cost charging.

- **Performance** data provides information for performance analysis, performance tuning, and includes records of specific events in the system.
  
  When you start a Performance trace, specify what you want to report, for example, I/O only or SQL only.

- **Audit** data provides information about DB2 security controls. You can use the data to ensure that data access is allowed only for authorized purposes.

  Tables are the only auditable objects. To audit a table, include the AUDIT clause in the CREATE TABLE or ALTER TABLE statement.

Statistics and Accounting data are used for the continuous or periodic monitoring of DB2, whereas Performance data is usually recorded only when you need to examine specific performance problems. Audit data is collected to monitor access to data.

The following table lists trace types, trace classes, and IFCIDs relevant to reports. Use the table to see which DB2 trace data is used as input for each report set.
**Table 3. Input for OMEGAMON XE for DB2 PE report sets**

<table>
<thead>
<tr>
<th>Report set</th>
<th>DB2 trace type</th>
<th>DB2 trace class</th>
<th>Description of DB2 trace class</th>
<th>DB2 IFCIDs used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting</td>
<td>Accounting</td>
<td>1</td>
<td>Accounting data</td>
<td>3, 239</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>In DB2 time</td>
<td>Additional information for 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Wait time in DB2</td>
<td>Additional information for 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>Time spent processing IFI requests</td>
<td>Additional information for 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>Package information - in DB2 time</td>
<td>Additional information for 3, 239</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>Package information - wait time in DB2</td>
<td>Additional information for 3, 239</td>
</tr>
<tr>
<td>Audit</td>
<td>Audit</td>
<td>1</td>
<td>Authorization failures</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Explicit GRANT or REVOKE</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>CREATE, ALTER, and DROP operations against audited tables</td>
<td>105, 107, 142</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>First change of audited object</td>
<td>105, 107, 143</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>First read of audited object</td>
<td>105, 107, 144</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>SQL statement at bind</td>
<td>105, 107, 145</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>Change in authorization for audited object</td>
<td>55, 83, 87, 169, 319</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>Utility access to any object</td>
<td>24, 105, 107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>Established trusted connections</td>
<td>269, 270</td>
</tr>
<tr>
<td>I/O Activity</td>
<td>Performance</td>
<td>4</td>
<td>Buffer manager I/O and EDM pool requests</td>
<td>6, 7, 8, 9, 10, 29, 30, 107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>Log manager</td>
<td>34, 35, 36, 37, 38, 39, 40, 41, 114, 115, 116, 119, 120</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21</td>
<td>Data sharing</td>
<td>107, 255</td>
</tr>
<tr>
<td>Locking</td>
<td>Statistics</td>
<td>3</td>
<td>Deadlock and timeout information</td>
<td>172, 196</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
<td>4</td>
<td>Buffer manager I/O and EDM pool requests</td>
<td>226, 227</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>Locking information</td>
<td>20, 44, 45, 172, 196, 213, 214, 218</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>Detailed locking information</td>
<td>21, 105, 107, 223</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17</td>
<td>Drain and claim</td>
<td>211, 212, 213, 214, 215, 216</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>Data sharing</td>
<td>251, 257</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21</td>
<td>Data sharing</td>
<td>259</td>
</tr>
<tr>
<td>Record Trace</td>
<td>All</td>
<td>All</td>
<td>All types, classes, and IFCIDs can be used as input</td>
<td>All</td>
</tr>
</tbody>
</table>
### Table 3. Input for OMEGAMON XE for DB2 PE report sets (continued)

<table>
<thead>
<tr>
<th>Report set</th>
<th>DB2 trace type</th>
<th>DB2 trace class</th>
<th>Description of DB2 trace class</th>
<th>DB2 IFCIDs used</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Activity</td>
<td>Accounting</td>
<td>1</td>
<td>Accounting data</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>In DB2 time</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Suspensions</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>IFI and data capture events</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>Package information - in DB2</td>
<td>239</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>Package information - wait time in DB2</td>
<td>239</td>
</tr>
<tr>
<td>SQL Activity</td>
<td>Performance</td>
<td>2</td>
<td>Subsystem-related events</td>
<td>68, 69, 70, 71, 72, 73, 74, 75, 84, 85, 86, 87, 88, 89, 106, 174, 175</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>SQL-related events</td>
<td>22, 53, 55, 58, 59, 60, 61, 62, 63, 64, 65, 66, 92, 95, 96, 97, 177, 233, 237, 272, 273</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>Buffer manager I/O and EDM pool requests</td>
<td>6, 7, 8, 9, 226, 227</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>Locking information</td>
<td>20, 44, 45, 213, 214, 218</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>Data Manager detail</td>
<td>15, 16, 17, 18, 106, 125, 221, 222, 231, 305, 325</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>Sort detail</td>
<td>28, 95, 96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>Autobind</td>
<td>105, 106, 107, 108, 109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>Edit and validation exits</td>
<td>11, 12, 19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>Distributed activity</td>
<td>157, 159, 160, 162, 163, 183</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17</td>
<td>Drain and claim detail</td>
<td>213, 214, 215, 216</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30, 31, 32</td>
<td>Installation-defined trace classes</td>
<td>188, 324</td>
</tr>
<tr>
<td>Statistics</td>
<td>Statistics</td>
<td>1</td>
<td>Statistics data</td>
<td>1, 2, 106, 202</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>Storage manager pool summary statistics</td>
<td>225</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>Buffer pool data set statistics</td>
<td>199</td>
</tr>
<tr>
<td>System parameters</td>
<td>Performance</td>
<td>Any</td>
<td>These IFCIDs are available in all trace classes</td>
<td>106, 201, 202, 256</td>
</tr>
<tr>
<td></td>
<td>Statistics</td>
<td>5</td>
<td>Data sharing global information</td>
<td>230</td>
</tr>
</tbody>
</table>
### Table 3. Input for OMEGAMON XE for DB2 PE report sets (continued)

<table>
<thead>
<tr>
<th>Report set</th>
<th>DB2 trace type</th>
<th>DB2 trace class</th>
<th>Description of DB2 trace class</th>
<th>DB2 IFCIDs used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility</td>
<td>Accounting</td>
<td>1</td>
<td>Accounting data</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
<td>3</td>
<td>SQL-related events</td>
<td>22, 63, 177</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>Buffer manager I/O and EDM pool requests</td>
<td>6, 7, 8, 9, 226, 227</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>Locking information</td>
<td>20, 44, 45, 213, 214, 218</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>Bind and utilities</td>
<td>23, 24, 25, 108, 109, 110, 111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>Edit and validation exits</td>
<td>11, 12, 19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>Distributed activity</td>
<td>183</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17</td>
<td>Drain and claim detail</td>
<td>213, 214, 215, 216</td>
</tr>
</tbody>
</table>

OMEGAMON XE for DB2 PE gathers input for Explain by connecting to an active DB2 subsystem.

If you want to see a count of the input trace records used in a job, you can produce an IFCID frequency distribution log.

### Collecting report data with the Collect Report Data Batch program

The new Collect Report Data Batch program allows for the collection of report data and postprocessing of collected report data by means of a batch job. This method provides a flexible way to automate the collection and postprocessing, thereby keeping the resource overhead minimized.

You can collect report data from a DB2 subsystem, a specified member of a data sharing group, or all members of a data sharing group. Collected report data is stored in a sequential data set or a Generation Data Group (GDG).

If a GDG is used, a user-specified job can be started to process collected report data automatically each time after a switch to a subsequent Generation Data Set (GDS) occurs. This method can be used to continuously collect trace data from DB2 and to automate the postprocessing. For example, the contents of the GDG generations (the Generation Data Sets) can be used to automatically generate reports or to load collected data into a Performance Database.

This method of collecting and postprocessing report data can be used as alternative to using System Management Facility (SMF) or Generalized Trace Facility (GTF) data sets if more flexibility regarding data collection parameters and error handling and less resource overhead are required.

The commands and parameters that determine the collection and postprocessing of data must be specified in the SYSIN DD statement of the respective JCL, as outlined in the following syntax diagram. A sample batch job can be found in data set `prefix.TKO2SAMP(FPEZCRDJ)`.
SSID
Specifies the DB2 subsystem ID ssid.

DSGLOBAL
Specifies whether to collect report data from all members of a data sharing group.
N  Specifies not to collect report data from all members. This is the default.
Y  Specifies to collect report data from all members. This setting is ignored if the specified subsystem ID ssid is not a member of a data sharing group.

DSMEMBER
Specifies the data sharing group member dsMbr from which to collect report data. dsMbr can be located on a different LPAR.

GDGDATASET
Is optional. If specified, it determines that the trace data collection will not terminate after a stop criteria is met, but the output data set will be switched to the next generation of the Generation Data Group (GDG), which means, to a subsequent Generation Data Set (GDS). Specify the name of an existing GDG as gdg (the GDG name, not a GDS name).
If specified, the JCL does not require an OUTPUT DD statement.
Use the PURGE operator command to stop CRD collection.

JOB
Is optional. If specified, job is executed after each switch to a subsequent GDS. job must be the name of a started job.
This can be used to automatically process the data in a GDS, for example, loading it into a Performance Database or generating reports.
If you require the actual name of the Generation Data Set (GDS) that was in use before the switch, you can specify the JCL symbol &GDGDSN in the job’s command stream, as in the following example:

//INPUTDD DD DSN=&GDGDSN,DISP=...

This method of using the JCL symbol ensures that the job is synchronized with the proper generation of the GDG, in case new generations are created faster than JES submits the specified job.

VOLUME
Is optional. A volume must be specified if a specified GDGDATASET is not managed by SMS.
UNIT
Is optional. A unit must be specified if a specified GDGDATASET is not
managed by SMS.

OPBUFSIZE
Specifies the size (in KB) of the Online Performance (OP) buffer that is used to
buffer collected data.

2048
Is the default size.

opSize
Can be up to 16384 KB.

ELAPSEDTIME
Specifies one (of multiple) stop criteria for the data collect task.

60
Is the default duration (in seconds).

secs
Specifies the duration (in seconds).

RECCOLLECTED
Specifies one (of multiple) stop criteria for the data collect task.

recs
Specifies the maximum number of IFCID records to collect.

TERMTHREADS
Specifies one (of multiple) stop criteria for the data collect task.

thds
Specifies the number of terminated threads after which data collection
stops.

TRACE
Specifies the DB2 START TRACE command to collect data. The command can
start with a preceding dash (-). Do not use the DEST or SCOPE subcommands
unless a specific OP buffer should be used.

The following excerpt from a batch job shows how two START TRACE
commands are used to collect different trace classes and IFCIDs:

```
...  
TRACE('START TRACE(PERFM) CLASS(30) IFCID(314) TDATA(CORRELATION,CPU)')
TRACE('START TRACE(AUDIT) CLASS(31) IFCID(140,83)')
```
The near-term history data collection options keyword WRITEOPTION in data set member RKD2PAR(COPT<ssid>) should have the value VSAM,SEQ.

On the Near-Term History Data Collection Options panel you should see Writeoption=VSAM,SEQ. This panel also lists the currently used data sets where collected data is stored.

For more information, see the Configuration Tool online help and Monitoring Performance from the OMEGAMON Classic Interface.

Creating reports with OMEGAMON XE for DB2 PE commands

OMEGAMON XE for DB2 PE has a set of commands to create reports.

Each command can use subcommands and subcommand options to control the period reported, what is included or excluded from the report, or which users are reported. The OMEGAMON XE for DB2 PE command stream is contained in a batch job, together with JCL statements. Typically, the batch job contains the following elements:
• Setup information
• General filters and controls
• Commands
• Subcommands and subcommand options

This topic assumes that the OMEGAMON XE for DB2 PE command stream in a JCL is created manually or a copy of an existing JCL is modified. You can also use the Interactive Report Facility (IRF) to specify reports interactively. The IRF then composes the command stream according to your specifications and executes the command stream in foreground or background mode. A subsequent topic describes the use of the IRF in more detail.

The following topics provide additional information:
• "Setup information for OMEGAMON XE for DB2 PE data sets"
• "General filters and controls" on page 33
• "Report set commands" on page 34
• "Subcommands and subcommand options" on page 34
• "Example of an OMEGAMON XE for DB2 PE command string" on page 35

Setup information for OMEGAMON XE for DB2 PE data sets

The setup information specifies the input and output data sets for your reports.

ddnames of general data sets

The following list shows the data definition names, also known as ddnames, for required and optional data sets.

The ddnames for the required data sets are:

**STEPLIB**
The data set that contains OMEGAMON XE for DB2 PE programs. You must specify this fully qualified data set name.

**SYSPIN**
The data set that contains OMEGAMON XE for DB2 PE command string. If the command string is contained in the JCL, specify an asterisk (*), otherwise specify the fully qualified data set name.
INPUTDD
The data set that contains the DB2 trace data to be used as input for OMEGAMON XE for DB2 PE reports. You must specify this fully qualified data set name.

The ddnames for the optional data sets are:

DPMLOG
The data set where processing messages are written.

SYSOUT
The data set where messages about sorting are written.

DPMPARMS
The data set that contains information about changes that you have made to OMEGAMON XE for DB2 PE standard processing settings. The things you can tailor are:
- Report layouts
- Time zone specifications (member LOCDATA)
- Correlation translation information (member CORRDATA)
- Exception field descriptions (member EXCHANGE)
- Definition of the main packages used in reporting (MAINPACK)

For more information, see "Customizing OMEGAMON XE for DB2 PE functions" on page 198.

DPMOUTDD
The output data set where OMEGAMON XE for DB2 PE writes formatted data. Specify a ddname for this data set only if you want to produce more reports from the same data later.

JOBSUMDD
The data set where information about OMEGAMON XE for DB2 PE processing is written. It contains the IFCID frequency distribution log and the job summary log.

DISTDD
The data set where output from the DISTRIBUTE command is written.

JSSRSDD
The data set where job summary data is written when a SAVE subcommand is processed.

The following three data sets are used for exception processing. For more information, see "Using Statistics and Accounting reports to identify exceptions" on page 58.

EXCPTDD
The data set where exception thresholds are stored. This data set is required for all exception processing.

EXTRCDD1
The data set where the Exception Log data is written.

EXFILDD1
The data set where the Exception Log File data is written.

ddnames of report data sets
Output from OMEGAMON XE for DB2 PE report set processing is written to the report data sets. You need to specify a data set for the SAVE, RESTORE, or FILE output corresponding to the particular report set you are requesting.
The default ddnames for these data sets all start with a two-letter prefix that indicates the report set. The following table shows the report set and the corresponding prefix:

*Table 4. Two-letter prefixes for OMEGAMON XE for DB2 PE report sets*

<table>
<thead>
<tr>
<th>Report set</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting</td>
<td>AC</td>
</tr>
<tr>
<td>Audit</td>
<td>AU</td>
</tr>
<tr>
<td>I/O Activity</td>
<td>IO</td>
</tr>
<tr>
<td>Locking</td>
<td>LO</td>
</tr>
<tr>
<td>Record Trace</td>
<td>RT</td>
</tr>
<tr>
<td>SQL Activity</td>
<td>SQ</td>
</tr>
<tr>
<td>Statistics</td>
<td>ST</td>
</tr>
<tr>
<td>System Parameters</td>
<td>SY</td>
</tr>
<tr>
<td>Utility Activity</td>
<td>UT</td>
</tr>
</tbody>
</table>

The following list shows the default ddnames for the report data sets, where cc is the prefix.

- **ccRPTDD**
  The data set where report output is written.

- **ccTRCDD1**
  The data set where trace output is written. If you generate more than one trace in the same job step, the second trace is written to ccTRCDD2, the third to ccTRCDD3, the fourth to ccTRCDD4, and the fifth to ccTRCDD5.

- **ccFILDD1**
  The data set where output from the FILE subcommand is written.

- **ccSAVDD**
  The data set where data is stored using the SAVE subcommand.

- **ccRSTDD**
  The data set from where data is read using the RESTORE subcommand.

- **ccWORK**
  The data set where output from the REDUCE subcommand is written. Normally this is a temporary data set that OMEGAMON XE for DB2 PE automatically creates and deletes. If you want to control the placement or size for this data set, specify a ddname.

The Explain report set does not use subcommands. Therefore, you need to specify only one data set. The default ddname for this data set is:

**EXPLAIN**
The data set where output from the EXPLAIN command is written.

**General filters and controls**

OMEGAMON XE for DB2 PE has a set of auxiliary commands to streamline the generation of reports.

These commands are shared by the various report sets.

**CASE** Use this command to accept entries in uppercase or lowercase characters. Specify CASE (SENSITIVE) before any other command if you want the
following commands to differentiate between uppercase and lowercase entries. If you do not specify the CASE command, or if you specify CASE (ANY), lowercase characters are converted to uppercase characters.

**FIELD** Use this command to include fields that match a comparison value in Record Trace reports.

**GLOBAL**

Use this command to identify the users, plans, or the period in time you want to investigate.

- You can filter the input data by specifying the start and end times of the data to be reported. This is done by using the FROM and TO subcommand options.
- You can filter the input data by specifying the identifiers for which you want data to be reported. This is done by using the INCLUDE and EXCLUDE subcommand options.

These values specified with the GLOBAL command are used as defaults in the subcommands.

**GROUP**

Use the GROUP command to define a group of OMEGAMON XE for DB2 PE identifier values for use in reporting. See “Group data” on page 195.

**LIST**

Use the LIST command to define a list of OMEGAMON XE for DB2 PE identifier values for use in reporting. See “Use lists” on page 196.

**Report set commands**

The report set commands specify the report set that you want to use. All report set commands have default subcommands, so you do not need to specify any subcommands.

The default subcommands usually produce the shortest report in the report set. See “Report sets” on page 5 for an overview of the different report sets.

Table 5. Report sets and report set commands

<table>
<thead>
<tr>
<th>Report set</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics report set</td>
<td>STATISTICS</td>
</tr>
<tr>
<td>Accounting report set</td>
<td>ACCOUNTING</td>
</tr>
<tr>
<td>Explain report set</td>
<td>EXPLAIN</td>
</tr>
<tr>
<td>SQL Activity report set</td>
<td>SQLACTIVITY</td>
</tr>
<tr>
<td>System Parameter report set</td>
<td>SYSPARMS</td>
</tr>
<tr>
<td>Utility Activity report set</td>
<td>UTILITY</td>
</tr>
<tr>
<td>Locking report set</td>
<td>LOCKING</td>
</tr>
<tr>
<td>I/O Activity report set</td>
<td>IOACTIVITY</td>
</tr>
<tr>
<td>Record Trace report set</td>
<td>RECTRACE</td>
</tr>
<tr>
<td>Audit report set</td>
<td>AUDIT</td>
</tr>
</tbody>
</table>

**Subcommands and subcommand options**

You can use subcommands and subcommand options to specify how you want the data to be presented.
REPORT
Use this subcommand to generate reports. In reports, data is summarized by OMEGAMON XE for DB2 PE identifiers, such as the primary authorization ID or the plan name.

Use the LAYOUT or LEVEL subcommand options of REPORT as appropriate to specify the amount of detail you want in the report. Use the ORDER subcommand option to specify how you want the data to be summarized. Use the EXCEPTION subcommand option to produce reports that contain only values outside user-specified limits.

TRACE
Use this subcommand to produce listings that show individual DB2 events, usually in the order of occurrence.

Use the LAYOUT or LEVEL subcommand option of TRACE as appropriate to specify the amount of detail you want in the trace. Use the EXCEPTION subcommand option to produce traces that contain only values outside user-specified limits.

FILE
Use this subcommand to store data about individual DB2 events in data sets that can be used with the DB2 LOAD utility.

Use the EXCEPTION subcommand option to produce data sets that contain only values outside user-specified limits.

REDUCE
Use this subcommand to aggregate Statistics and Accounting DB2 events. REDUCE consolidates DB2 events with the same OMEGAMON XE for DB2 PE identifiers. You can save the reduced data by using the SAVE command.

Use the INTERVAL and BOUNDARY subcommand options of REDUCE to specify how the data is consolidated. INTERVAL specifies the time range at which records are consolidated. BOUNDARY specifies the start time of the INTERVAL.

If SAVE or processing by INTERVAL is not required, you can omit REDUCE.

SAVE
Use this subcommand to save reduced data. You can use the saved data in later reporting. You can also convert the data set into a sequential data set that can be loaded into DB2 tables using the Save-File utility.

RESTORE
Use this subcommand to include previously saved data.

Example of an OMEGAMON XE for DB2 PE command string
This example shows a JCL that produces a short Accounting report and a long Statistics trace.

The text that follows describes the contents of the JCL and shows an example output from this JCL.
The first two lines are user and environment settings. Change these settings according to your operating environment.

Line 3 contains the call to OMEGAMON XE for DB2 PE. The DATEFORMAT parameter determines the mm/dd/yy format (which is the default format, so it is not strictly necessary to be declared in the job stream).

The STEPLIB statement (line 4) must be modified to point to your OMEGAMON XE for DB2 PE installation.

The SYSIN statement (line 5) indicates that the command string is contained within this JCL.

The INPUTDD statement (line 6) specifies the input data set that contains DB2 trace data.

The GLOBAL command (line 8) restricts the data that is passed from the input data set to the ACCOUNTING and STATISTICS commands. Only data between the specified FROM and TO dates and times is made available to the commands.

The ACCOUNTING command (line 11) produces an Accounting report. The REPORT subcommand (line 12) is not strictly necessary because REPORT is the default subcommand for ACCOUNTING. The FROM and TO subcommand options (lines 13 and 14) restrict the period of the Accounting report to 30 minutes. When you use these subcommand options, ensure that the period is within the period specified by the GLOBAL command. Otherwise, no report is generated.

The STATISTICS command (line 15), its TRACE subcommand (line 16), and its LAYOUT subcommand option (line 17) produce a long Statistics trace.

You need to include the EXEC statement (line 18) to generate the reports. Otherwise, no report is generated.
Because no explicit output data sets are specified in this example, the reports will be written to ACRPTDD for the Accounting report and STTRCDD1 for the Statistics trace, which are the default ddnames.

To submit the job, type `SUBMIT` on the command line and press Enter.

The syntax of your JCL is checked and written to the DPMLOG data set, together with any information, warning, or error messages raised.

**Figure 14. Submitting the job**

The syntax of your JCL is checked and written to the DPMLOG data set, together with any information, warning, or error messages raised.

**Figure 15 shows the Accounting report produced in the previous JCL.**

**Creating reports with the Interactive Report Facility (IRF)**

The Interactive Report Facility (IRF) provides a dialog, though a series of panels, that guides you through the specifications for a desired report.

After you specified a report, your specifications are validated and the IRF automatically generates a batch report command stream with JCL statements and the commands, subcommands, options and keywords that match your selections for the requested reports. This job can be executed in foreground or background mode to produce the report.

The IRF is invoked from the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS main menu by selecting option 1 (Create and execute reporting commands).
Selecting reports and reports details

In the Interactive Report Selections panel, you select which reports you want to create, the functions to apply to the selected reports, and additional functions to be reflected in the batch report command stream.

About this task

When you invoke option 1 (Create and execute reporting commands) from the main menu, the following panel is displayed.

![Interactive Report Selections panel]

The panel provides a matrix for selecting report sets (Accounting, Statistics, and so on) and the functions (Reduce, Report, and so on) to be applied to the selected report sets. Multiple functions can be applied to selected report sets, but some functions are inapplicable to some report sets, as shown by the matrix.

- Use forward slashes (/) to select combinations of report sets and functions (for example, an Accounting report).
- Greater than symbols (>) indicate previously made selections (for example, a Statistics report).
- To correct a selection, overtype the symbol with a blank.

The selections of Additional Functions are treated in the same manner.

The command line accepts the following commands to facilitate and complete the report selections:

- The INCLUDE and EXCLUDE commands filter data on a report set level or at a global level.
- The GROUP command defines a named group of OMEGAMON XE for DB2 PE identifiers. The group name can be used when you request reports.
- The LIST command defines a named group of OMEGAMON XE for DB2 PE identifiers. The list name can be used in INCLUDE and EXCLUDE commands instead of individually entering each list member.
• The BROWSE command can be used to review the batch report command stream that was generated based on your current selections.
• The SAVE and RECALL commands save current selections or recall previously saved selections.
• The OPTIONS command sets or changes the default options of your current IRF session, such as foreground or background processing and the ddnames to be used.
• The RESET command clears all input fields.

When your specifications for the desired reports are complete, press F5 (Compose).

Specifying default options by using the OPTIONS command
You can change the defaults by using the OPTIONS command before you press F5 (Compose).

About this task
If no defaults were created before, the following steps are automatically invoked after you pressed F5 (Compose). Otherwise, these steps are skipped and the default options are applied to the generation of the batch report command stream.

The Execution Mode panel is displayed for selecting foreground or background processing.

Depending on your selection, the Foreground DDname Selections or Background DDname Selections panel is displayed.

For each ddname, enter one of the following to update the data set information:
• Data set name
  The name of the input data set or the name of the data set where output is directed to. If the name is not enclosed in apostrophes, the TSO prefix is added to it.
• Extended information for a ddname
  You can specify additional parameters needed for the DD statement in JCL syntax (background), or for the TSO/E ALLOC command (foreground).
• Asterisk (*)
  The output is directed to the terminal (foreground only).
• Blank
  The ddname is not used, or it is dynamically allocated.

Mandatory input fields are marked by an asterisk (*) in the Required field.

Press Enter to validate the entries.

Composing and executing the batch report command stream
After all defaults are specified and F5 (Compose) was pressed, your selections and specifications are validated and the job stream is generated.

About this task
If foreground processing was selected, the job is executed immediately. Your terminal remains busy until the job is completed.
If background processing was selected, the Job Processing Selections panel is displayed. In this panel, you can:

- Browse the generated job stream.
- Apply changes to the job stream before you submit it.
- Store the job stream for future use (after the job has been saved, it can be edited with any standard editor, for example, ISPF/PDF EDIT).
- Submit the job.
- Specify the required information for the JOB statement.

**Saving and recalling selections**

The SAVE command saves all report set, reporting command, and ddname selections and specifications that you have made so far into a partitioned data set.

**About this task**

You can also save incomplete selections and use these selections as a template for a specific type of report. The SAVE command displays the Save Selections panel, where you can specify the data set and member name in which selections are to be saved. This data set must exist and must be defined with the following attributes:

- **RECFM**: FB
- **LRECL**: 80
- **BLKSIZE**: 6160

**Directory blocks**

Depends on how many members you want to save.

The next time you want to produce the same or a similar command stream, type **RECALL** on the command line of one of the IRF panels. The RECALL command displays the Recall Selections panel, where you can specify the data set and member in which the previous selections have been saved. At this time, you can still modify or complete the recalled selections according to your needs. If you want to use the recalled selections unchanged, enter **COMPOSE** on the command line to generate the JCL and the command stream.

**Example of producing an Accounting report**

The following steps show how you can produce a short Accounting report.

**About this task**

Comprehensive help information is available on all IRF panels.

1. Start the IRF by invoking option 1 (Create and execute reporting commands) from the main menu. The Interactive Report Selections panel is displayed:
2. Type `RESET` on the command line to delete all previously made selections from the panel.

3. Select an Accounting report by entering a forward slash (`/`) on the corresponding line, as shown in the previous figure.

4. Type `OPTIONS` on the command line. The Session Options window is displayed, overlaying the Interactive Report Selections panel:

```
Interactive Report Selections
Select functions as required, then press Enter.

Report Set       --------------- Function ---------------
                  Reduce Report Trace File Save Restore

Additional Functions
Global Processing
Frequency Distribution
System Parameters
Exception log
Explain

Command ===> _________________________________________________________________
F1=Help F2=Split F3=Exit F5=Compose F6=Browse F9=Swap
F10=Global F11=Inclexcl F12=Cancel
```

Figure 17. Interactive Report Selections panel

5. In the Execution mode field, type `1` to select background processing and press Enter. The Interactive Report Selections panel is displayed again.

```
Interactive Report Selections

S R Update fields as required, then press Enter.
A R Confirmation display . . . . . . . . . . . . . 1 1=yes 2=no
S S Initial menu choice . . . . . . . . . . . . . 1-6 or blank
S L Execution mode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1=Background
I A 2=Foreground
U A 3=Prompt
R A DMPPARMS data set...

Global Processing
Frequency Distribution
System Parameters
Exception log
Explain

Command ===> options
F1=Help F2=Split F3=Exit F5=Compose F6=Browse F9=Swap
F10=Global F11=Inclexcl F12=Cancel
```

Figure 18. Interactive Report Selections panel - Session Options window
6. Press Enter. The Accounting Report Selections panel is displayed:

```
Accounting REPORT Selections
Select one or more reports, then press Enter. Overtype with space to delete any report. Request EXIT when complete.

<table>
<thead>
<tr>
<th>DDname</th>
<th>User Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>My Accounting report short_____</td>
</tr>
</tbody>
</table>

Command ===> F1=Help F2=Split F3=Exit F6=Browse F7=Up F8=Down F9=Swap F12=Cancel
```

**Figure 19. Accounting Report Selections panel**

7. Type a forward slash (/) in the action field and give the report an appropriate name in the User Comment field, as shown in the previous figure. You do not need to specify a ddname. The default is used.

8. Press Enter. The Accounting Report panel is displayed:

```
Accounting REPORT
Update fields as required, then press Enter.

User comment ........ My Accounting report short_____  
DDname ............ ACRPTDD_ 
Scope ............. 1 =member 2 =group 
Layout ............ short_ User-tailored Report Format 
Exception ........ 2 =yes 2 =no 
Select to change values or overtype with space to use default.

Top Entries
Order Selections

YY MM DD HH MM SS TH 
Report from .......... 
Report to ..........: 

Command ===> F1=Help F2=Split F3=Exit F6=Browse F9=Swap F10=Global F11=Inclexcl F12=Cancel
```

**Figure 20. Accounting Report panel**
Note the default ddname. In this panel you can refine the desired Accounting report. Here, member-scope reporting and a short report are specified.

9. Press Enter to return to the Accounting Report Selections panel.

10. Press F3 (Exit) to return to the Interactive Report Selections panel. You have completed the specifications for the commands, subcommands, options and keywords that are required to generate the example Accounting report.

11. You can use F6 (Browse) to view the command stream. This displays the following panel:

```
ACCOUNTING
REPORT /*My Accounting report short*/
  DDNAME(ACRPTDD)  
  LAYOUT(SHORT)  
  SCOPE(MEMBER)  
EXEC
```

Figure 21. Accounting report command stream

12. On the Interactive Report Selections panel, press F5 (Compose). This generates the JCL and command stream, which you can browse, edit, store, or execute. The Background DDname Selections panel is displayed (because of your previous execution mode selection):

```
DDname  Data Set Information  Required
INPUTDD  'HECK.ACC.INPUT.SMF'  *
EXCPTDD
EXTRCDD1
EXFILDD1
ACRPTDD
DMLLOG  SYSOUT=A
DPMOUTDD
DPMPARMS
JOBSUMDD
JSSRSDD
SYSOUT  SYSOUT=A
SYSDUMP
```

Figure 22. Background DDname Selections panel

13. Enter at least the required information, marked by asterisks (*). Usually, you need to specify the input data set that contains the data to be reported. In this example, ACC.INPUT.SMF is entered in the INPUTDD line, which resolves to 'HECK.ACC.INPUT.SMF' (the TSO prefix is added) when you press Enter.
The list of ddnames also shows the default ddname for Accounting reports (ACRPTDD). Output data sets are dynamically allocated, if not specified explicitly. If you want to become familiar with the IRF and do not have real input data available, you can use the sample data set DB2PM.V400.DPMIN40. If this data set is not available, press F1 (Help) and F5 (Exhelp) for current information about the sample data set.

14. Press Enter. The Job Processing Selections panel is displayed:

```
Job Processing Selections
Update the job statements as required, then select one of the following.
1. Browse the generated job stream
2. Edit the generated job stream
3. Store the job stream for future use
4. Submit the job stream for background execution

Job statement information

Command ===> __________________________________________________________
F1=Help   F2=Split   F3=Exit   F9=Swap   F12=Cancel
```

Figure 23. Job Processing Selections panel

15. You can now browse, edit, store, or submit the generated job stream. For example, select option 1 (Browse the generated job stream) to display a panel with the following job stream:

```
**Figure 24. Interactive Report Facility (IRF) - example of generated job stream**

16. To submit the job, specify appropriate job statement information at the bottom half of the Job Processing Selections panel and press Enter. You are prompted if this information is missing.
You should now see a message that indicates whether the job was successfully submitted.
Chapter 3. Using OMEGAMON XE for DB2 PE reports

The following information describes how to use and interpret OMEGAMON XE for DB2 PE reports.

Using Statistics traces to get system overview information

The best way to get started with OMEGAMON XE for DB2 PE is to create a long Statistics trace report to get an overview of your DB2 system.

To do this, you need to:
- Run a DB2 trace to collect DB2 Statistics data during a 24-hour period.
- Create a Statistics trace report.
- Interpret the report and adjust the system.

The reason for creating a trace report is that the 24-hour period is reported as a single interval. A normal report creates a record for each Statistics interval, which could result in an output of more than 1,200 pages.

Choose a day that will have a fairly typical workload, not a weekend, for example, or a public holiday when workload will probably be much reduced.

The following topics provide additional information:
- "Collecting Statistics data with a DB2 trace"
- "Creating a Statistics trace report"
- "Interpreting the Statistics trace report and applying rules of thumb" on page 48

Collecting Statistics data with a DB2 trace

Use the workstation client to configure and start a DB2 trace to collect DB2 Statistics data during a 24-hour period.

About this task

To do this, use the workstation client System Overview window.
1. From the menu Traces bar select Traces → Configurations to open the Trace Configuration window.
2. Click the Create button to create a new configuration.
3. Use the Name window to identify the trace configuration. You can give the configuration a meaningful name, specify the output data set, and specify whether the data set is overwritten or appended. Select the Overwrite radio button.
4. In the Data window, select the Statistics check box and ensure that all other check boxes are clear. The IFCIDs associated with this trace are displayed and highlighted. Only the highlighted IFCIDs are collected in the trace. Here you can, for example, choose not to include buffer pool statistics at data set level.
5. In the Start window select the Immediate start radio button. In the Stop window, select the Elapsed time check box and specify a time of 24:00:00.
6. Click OK to start the trace.

Creating a Statistics trace report

Use your ISPF editor to create a JCL job that creates a Statistics trace report.
About this task

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.

```jcl
EDIT ---- ANDREW.OMPE.JOBS.STTRALON------------------------- Columns 001 072
Command ===>

****** ********************************* TOP OF DATA *********************************
0001 /* JCL to produce a Statistics LONG trace report */
0002 //ANOMPES JOB (TTS1,YUS7), 'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //SYSIN DD *
0007 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE1,DISP=SHR
0008 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.STTRCA01,DISP=OVR
0009 //JOBSUMDD DD SYSOUT=A
0010 STATISTICS
0011 TRACE
0012 LAYOUT(LONG)
0013 FROM (07/16/08,00:01)
0014 TO (07/17/08,00:01)
0015 EXEC

****** ********************************* BOTTOM OF DATA *********************************
```

Figure 26. Specifying a job stream using ISPF/PDF editor

To submit the job, type SUBMIT on the command line and press Enter.

Interpreting the Statistics trace report and applying rules of thumb

You can use the Statistics trace report that you have produced in the previous steps for a quick assessment of the performance of your DB2 system.

A good way of looking at the data is to focus on a few key indicators and apply a few simple rules and ratios, known as rules of thumb (ROTs). Rules of thumb have been assembled by DB2 experts in the field over many years and are proven to be valuable criteria to measure the performance of applications in a DB2 system.

Rules of thumb (ROTs) are also implemented as samples in the Performance Warehouse. You can examine the ROT definitions and use the Performance Warehouse to evaluate your DB2 systems. For more information, see Monitoring Performance from the Workstation.

Getting an overview of your DB2 system performance

The Statistics trace report provides an overview of important areas, like DBM1 storage usage, data set activities, logging and locking activities, and thread management.

To get an overview of your DB2 system, you need to look at the following areas:

- **DBM1 storage**
  
  DBM1 storage is used for buffer pools, EDM pools, RID pools, and authorization caches. Because DBM1 storage is restricted to 2 GB, it is essential that the various pools and caches are optimized in this space.

- **Data set Open and Close activity**
  
  Applications that leave data sets open can affect other applications that must wait because too many data sets are already open.

- **Logging**
Logging activities interrupt normal DB2 operations. Logging problems will therefore affect applications and performance.

- **Locking**
  Locks occur in DB2 when applications have to wait for resources (concurrency problems).

- **ROWID**
  The ROWID block of the report shows the success of direct row access attempts.

- **Thread management**
  Thread optimization avoids applications from having to wait for free storage to allow thread creation.

**Buffer pool efficiency**

The buffer pool related report blocks of the Statistics trace report begin roughly at page 9 of the report. They are repeated for every active buffer pool.

The efficiency of your buffer pools directly affects the performance of your DB2 system. Well tuned buffer pools reduce the number of read and write operations from and to a hard disk drive, which in turn reduces application wait times.

**Tip:** OMEGAMON XE for DB2 PE includes the Buffer Pool Analyzer. This is a powerful tool that reports on the efficiency of your buffer pools and simulates your DB2 workload to arrive at an optimal buffer pool configuration for the system. For more information, see the *Buffer Pool Analyzer User's Guide*.

**Hit ratios**

The Buffer Pool Read block provides hit ratios that serve as indicators of the overall buffer pool efficiency.

The buffer pool hit ratio (BPOOL HIT RATIO (%)) shows the number of Getpage requests issued by applications and satisfied by the buffer pool, expressed as a percentage of all Getpage requests. This is a relative value that depends on the type of application. For example, applications that browse large amounts of noncontiguous data could cause the buffer pool hit ratio to drop near 0.

The hiperpool hit ratio (HPOOL HIT RATIO (%)) shows how the hiperpool is contributing to I/O avoidance. Compare this field with the hiperpool read/write ratio (HPOOL R/W RATIO (%)).

**Table 6. Hiperpool efficiency**

<table>
<thead>
<tr>
<th>HPOOL HIT RATIO (%)</th>
<th>HPOOL R/W RATIO (%)</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>High</td>
<td>Hiperpool is working efficiently.</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>Hiperpool use is ineffective.</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Hiperpool use is ineffective.</td>
</tr>
</tbody>
</table>

**Avoid page-ins**

The Buffer Pool Read block and the Buffer Pool Write block provides indications of potential performance problems that are caused by paging activities to a hard disk drive.

A buffer pool that is too large can also cause performance problems because it causes paging to a hard disk drive.
Look at the PAGE-INS REQUIRED FOR READ and PAGE-INS REQUIRED FOR WRITE fields in the Buffer Pool Read block and the Buffer Pool Write block of the report. These values should be zero, or close to zero. Higher values mean that the buffer pool size is over allocated. Consider reducing the buffer pool size.

**Optimize prefetch**

The Buffer Pool Read block provides indications of the efficiency of DB2's page prefetch activities.

DB2 uses prefetch to optimize queries. For SQL, a prefetch can read up to 32 pages from a hard disk drive. Prefetch can stop or be canceled when more than 90% of the pages in the buffer pool is unavailable. This can have a considerable effect on performance because scanned pages must be read synchronously from a hard disk drive.

Check the PREF.DISABLED-NO BUFFER and DM CRITICAL THRESHOLD fields in the Buffer Pool Read block of the report. If one or both field values are not zero or not close to zero:

- Review the Sequential Steal thresholds (VPSEQT and HPSEQT)
  
  These thresholds are percentages of the virtual buffer and the hiperpool that might be occupied by sequentially accessed pages. If these thresholds are too low, prefetch can be disabled. If these thresholds are too high, other thresholds can be reached too soon.

- Reduce the Deferred Write thresholds (VDWQT and DWQT)
  
  If the DB2 system is used mainly for high-use query systems reliant on prefetch, reducing these thresholds decreases the portion of the virtual buffer pool that is occupied by updated pages.

- Increase the size of the buffer pool
  
  Use the ALTER BUFFERPOOL command to increase the size of the buffer pool, which will give the system more space for prefetched pages.

**EDM pool failures**

The EDM pool stores cursor tables (CT), package tables (PT), and database descriptors (DBD).

Any pool full failures (FAILS DUE TO POOL FULL) indicate that the EDM pool is too small for the DB2 workload. The EDM pool should be big enough to store the cursor tables (CT), package tables (PT), and database descriptors (DBD). The ratios CT REQUESTS/CT NOT IN EDM, PT REQUESTS/CT NOT IN EDM, and DBD REQUESTS/CT NOT IN EDM show the EDM pool utilization. Generally, ratios of about 80% are adequate in most cases. If these values are significantly lower, reduce the size of the EDM pool and allocate the saved storage to the buffer pool, where the performance benefits are greater.

**Authorization problems**

DB2 performance can be affected by plans and packages that are waiting for authorization checks or by failures because the authorization cache is full.

Compare the number of authorization checks made for authorized plans (PLAN-AUTH SUCC) with the number of checks made that did not use the DB2 catalog (PLAN-AUTH SUCC-W/O CATALOG). A wide difference could indicate a first use of any plan by a given user ID after DB2 was started. Set the CACHESIZE parameter to a value greater than zero for all plans where EXECUTE privilege is not granted to public.
If the number of package authorization failures (AUTH UNSUCC-PKG-CACHE) is not zero, or close to zero, use the ZPARM ZPAC to increase the size of the package cache.

**RID List failures**
RID List failures occur if the RID pool size is constrained.

If the number of times that DB2 failed to allocate storage for RID list processing caused by storage constraints (TERMINATED-NO STORAGE) is not zero, or close to zero, you should reduce the storage size that is used by other structures.

Failures because RDS or Data Manager limits are exceeded (TERMINATED-EXCEED RDS LIMIT and TERMINATED-EXCEED DM LIMIT) are caused by inaccurate or incomplete RUNSTATS statistics or by optimizer errors. Add the clause OPTIMIZE FOR 1 ROW to the SQL statement to avoid RID list processing, or add the necessary columns to the index to make access INDEX ONLY.

If the number of RID list failures caused by storage limits (TERMINATED-EXCEED PROC LIMIT) is not zero, or close to zero, increase the RID pool size.

**Locking problems**
TIMEOUTS and DEADLOCKS, shown in the Locking Activity block of the Statistics trace report, should be close to zero.

If not, then if the number of LOCK REQUESTS is not significantly less than the total number of SQL DML requests, lock avoidance is not active. Set BIND CURRENT DATA to NO, declare the cursor with FOR UPDATE OF, and use the WHERE CURRENT OF cursor name with UPDATE and DELETE.

**Logging performance**
Logging can cause performance problems when the output buffer is inappropriate.

In the Log Activity block of the report, the READS SATISFIED-ARCH.LOG(%) field value should be low and the UNAVAILABLE OUTPUT LOG BUFF field value should be zero. If necessary, change the OUTBUFF parameter to increase the size of the output buffer.

**Data set Open and Close activities**
This report block of the Statistics trace report indicates whether enough data sets are allowed to be concurrently open.

The DSETS CLOSED-THRESH.REACHED field value should be close to zero. If not, use the ZPARM DSMAX to allow more open data sets. A reasonable figure is 8 000 - 10 000.

Set the CLOSERULE parameter of the CREATE TABLESPACE, CREATE INDEX, ALTER TABLESPACE, and ALTER INDEX commands to YES for table spaces and indexes that are used mainly by batch applications, and to NO for those that are used mainly online.

**ROWID efficiency**
This report block shows how efficient direct row access is used. Direct row access is very fast because DB2 does not need to use an index or table space scan to find the row.
The TABLE SPACE SCAN USED field value should be zero. Use the WHERE clause to cause an unsuccessful direct row access to revert to a matching index scan using the primary key:

```
WHERE rowidcol=:HVROWID AND pkcol=:HVPK
```

**Thread management**
The Subsystem Services block of the Statistics trace report indicates how thread queuing is performed.

The QUEUED AT THREAD CREATE field shows the number of allied and allied-distributed threads that were not immediately created because the maximum number of threads allowed in the system had been reached.

As a rule of thumb about 1% thread queuing is acceptable. When this is appreciably higher, increase the value of MAX USERS on the DB2 install panel DSNTIPE.

The combined maximum allowed for MAX USERS and MAX REMOTE ACTIVE cannot exceed 2000.

The DBAT QUEUED-MAXIMUM ACTIVE field in the Global DDF Activity block of the report shows the number of DBATs that had to wait because the maximum number of threads allowed in the system had been reached. This value should be zero, or close to zero. Increase the value of MAXDBAT to raise the number of current active DBATs allowed.

### Using an Accounting report to analyze resource activities

The Accounting report shows how threads and applications perform in DB2.

The Accounting report set consists of a report and a trace report. The Accounting report accumulates and groups Accounting records by OMEGAMON XE for DB2 PE identifiers, such as primary authorization or plan name. The Accounting trace report reports single threads. A trace report can be very long because it shows a single record for each active thread in the system.

A good first indicator for applications is the time used per thread by any particular user. A strategy for improving application performance is to identify the users with the most system time consumption and examine their most frequently used applications:
- Create a DB2 trace for input to a report
- Create a short Accounting report that identifies the users with the highest system time consumption.
- Interpret the report and identify users and plans that need attention
- Create an Accounting trace report for each problem user and plan
- Interpret the trace report and find remedial actions
- Apply rules of thumb in the Performance Warehouse

The following topics provide additional information:
- “Collecting Accounting data with a DB2 trace” on page 53
- “Creating a short Accounting report” on page 53
- “Interpreting the Accounting report and making changes” on page 53
- “Interpreting the Accounting trace report and applying rules of thumb” on page 55
Collecting Accounting data with a DB2 trace

Use the workstation client to configure and start a DB2 trace to collect DB2 Accounting data during a 24-hour period.

About this task

To do this, use the workstation Client System Overview window.
1. From the menu bar select Traces → Configurations to open the Trace Configuration window.
2. Click the Create button to create a new configuration.
3. Use the Name window to identify the trace configuration. You can give the configuration a meaningful name, specify the output data set, and specify whether the data set is overwritten or appended. Select the Overwrite radio button.
4. In the Data window, select the Accounting check box and ensure that all other check boxes are clear. The IFCIDs associated with this trace are displayed and highlighted. Only the highlighted IFCIDs are collected in the trace.
5. In the Start window select the Immediate start radio button. In the Stop window, select the Elapsed time check box and specify a time of 24:00:00.
6. Click OK to start the trace.

Creating a short Accounting report

Use your ISPF editor to create a JCL job that creates a short Accounting report.

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.

```
EDIT ---- ANDREW.OMPE.JOBS.ACREPSHO------------------------- Columns 001 072
Command ===> ________________________________________________ Scroll ===> CSR_
****** *********************************************** TOP OF DATA ***********************************************
0001 /* JCL to produce an accounting short report */
0002 //AWOMPEA JOB (TS1,YUS7), 'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //SYSIN DD *
0007 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE2,DISP=SHR
0008 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.ACRPT01,DISP=OVR
0009 //JOBSUMDD DD SYSOUT=A
0010 ACCOUNTING
0011 REPORT
0012 TOP(20)
0013 EXEC
****** *********************************************** BOTTOM OF DATA ***********************************************
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel
```

Figure 27. Specifying a job stream using ISPF/PDF editor

To submit the job, type SUBMIT on the command line and press Enter.

Interpreting the Accounting report and making changes

An example of a short Accounting report is evaluated and key fields are emphasized.

The output from your short report could look similar to the following report:
In this example, the application with the highest elapsed time is JUSTIN-JUSDOIT3.

There are a couple of points of interest for this user application that can give you an idea of what to look for in a more detailed report:

- Elapsed times and CPU times
- Large differences between class 1 and class 2 times
- Synchronous reads
- Lock suspensions

All of JUSTIN's applications in this example show similar symptoms. It is also possible that these applications could be affecting other applications. If these problems could be solved, general thread throughput could be improved.

Generate an Accounting trace report to have a closer look at JUSTIN's jobs. Use the same input data to create the report output. The JCL looks like this:
Interpreting the Accounting trace report and applying rules of thumb

Using an Accounting trace report for a specific problem analysis and applying expert rules of thumb.

The long Accounting trace report shows much information, probably much more than you need for a quick analysis. A good way of looking at the data is to focus on a few key indicators and apply a few simple rules and ratios, known as rules of thumb (ROTs). Rules of thumb have been assembled by DB2 experts in the field over many years and are proven to be valuable criteria to measure the performance of applications in a DB2 system.

Rules of thumb (ROTs) are also implemented as samples in the Performance Warehouse. You can examine the ROT definitions and use the Performance Warehouse to evaluate your DB2 systems. For more information, see Monitoring Performance from the Workstation.

The number of Getpages per SQL statement

Getpages per SQL statement indicate the number of pages that must be accessed for each data row written or read by an application.

Normally, when data rows are read randomly with a unique-key index, one page is read for each index level and one page is read for the data. Typically, the number of pages read to retrieve a data row would be less than six. Similarly, when pages are read sequentially, multiple rows can be retrieved for each page read.

A high Getpage/SQL ratio could be caused by:

- The use of table scan. This causes each page in the table to be accessed and scanned, even though only one row is returned.
- The use of nonmatching index scan. When this happens, one Getpage is made for each data page in the table and each index page in the associated index, even though only one row is returned.
- The use of an index with low cardinality. This can result in multiple page scans before a row is returned.
Comparison of class 1 and class 2 times
The ratio of class 1 and class 2 times indicate whether a potential problem lies in DB2.

A difference of roughly 10% in these times is normal.

If the class 2 elapsed time is less than half of the class 1 elapsed time, the problem is not with DB2. Possible causes include:
- OS/390
- CICS
- IMS
- Other programs

If the difference between the class 1 and class 2 elapsed times is somewhere between 10% and 50%, the problem lies somewhere in DB2. Verify the class 2 times.

Comparison of class 2 elapsed and class 2 CPU times
The ratio of class 2 elapsed and class 2 CPU times indicates a potential problem with SQL.

If the difference between the class 2 elapsed time and class 2 CPU times is between 10% and 50% (when CL2 CPU/CL2 ELAPSED is greater than 0.5), the problem probably lies with SQL. Use an EXPLAIN or SQL Activity report.

If the class 2 elapsed time is significantly greater (when CL2 CPU/CL2 ELAPSED is less than 0.5), the application is experiencing long wait times. Verify the class 3 suspension times.

Class 2 CPU time per SQL DML statement execution
Class 2 CPU time per SQL DML statement execution time helps to identify potential access path problems.

The time taken for an SQL DML statement to complete depends on the speed of your processor. Typically, for a 100 MIPS CPU, a DML statement should complete in less than one millisecond. Because some DML statements take more cycles to complete than others, the statements are weighted as follows: SELECT+INSERT+UPDATE+OPEN+(FETCH * 0.1)+(PREPARE * 5).

If you find that your SQL statement time is averaging well over the one millisecond, you can use the Workstation Online Monitor to identify the problem statements and use the drill-down capability to view these at statement level. If you suspect an access path problem, you can use Visual Explain to show the access path selected.

Not accounted time
Excessive time that is not recorded by DB2 indicate potential problems with overloaded servers.

This is time that is not recorded by DB2, either in class 2 time or in class 3 suspension time. This time should be below 25% of the total class 2 time. A higher percentage usually indicates an overloaded server, which could cause problems for online transactions. When this figure is consistently high, you should consider reducing the workload on the server or installing a bigger server.
Class 3 suspensions
Class 3 suspensions indicate potential problems with lock acquisitions, long wait times for synchronous I/O, and long service task switch times.

If the class 3 times are not high, the application wait times could be caused by CPU queuing or by OS/390 paging. This should be shown as class 2 not accounted time.

When evaluating class 3 suspensions, check the following information.

Commit interval
This is calculated as:

\[
\text{COMMIT INTERVAL} = \frac{\text{CLASS 2 ELAPSED TIME}}{\text{COMMITS+ROLLBACKS}}
\]

An interval of 1 to 5 seconds is acceptable. Higher values indicate problems with lock acquisition, especially in batch applications.

Class 3 synchronous I/O time per I/O event
Typically, this time should be about 20 to 30 milliseconds. A longer time indicates a problem because applications are waiting for synchronous I/O.

You can reduce I/O contention by increasing the size of your buffer pools or by reorganizing table and index spaces using REORG.

I/O problems are often system related. If a corresponding Statistics report also indicates problems, take the appropriate action. There could also be RMF™ problems. Verify the buffer pool data set statistics block of the report.

Service task switch
The accumulated wait time from switching synchronous execution units, by which DB2 switches from one execution unit to another.

The most common contributors to service task suspensions are:
- Wait for commit processing for updates (UPDATE COMMIT)
- Wait for OPEN/CLOSE service task (including HSM recall)
- Wait for SYSLGRNG recording service task
- Wait for data set extend/delete/define service task (EXT/DEL/DEF)
- Wait for other service tasks (OTHER SERVICE)

The significant fields here are the times used by:
- UPDATE COMMIT
  This is the average service time associated with a DB2 commit, abort, or deallocation. Normally this value should be below 12 milliseconds. If this value is significantly higher, verify the application or raise the interval between commits.
- OPEN/CLOSE
  If the average time for an OPEN/CLOSE exceeds 150 milliseconds, DS MAX can be too low or too many data sets are open in the system. Adjust the DS MAX threshold. A figure of between 8000 and 10000 is a good working size in most cases.
  Ensure that applications use the correct QUERYTYPE. Set the CLOSERULE parameter of the CREATE TABLESPACE, CREATE INDEX, ALTER TABLESPACE, and ALTER INDEX commands to YES for table spaces and indexes used mainly by batch and NO for those used mainly online.
Using Statistics and Accounting reports to identify exceptions

Exception reporting is a very effective way to identify performance problems. Exception reporting identifies DB2 threads and Statistics intervals with fields that contain values outside defined thresholds. This helps you manage performance objectives by highlighting problems in the DB2 subsystem, such as applications that are experiencing exceptional conditions or DB2 subsystem conditions that are causing thread performance problems.

You should run Accounting and Statistics exception reports as part of your regular monitoring. Exception reporting is also available in the Online Monitor.

The thresholds are set in the Exception Threshold data set. You can define exception thresholds for a number of fields on a plan or program basis (commonly known as accounting or thread fields) and on a system basis (known as statistics fields). When you request exception reporting, the input data is checked against these values. Only records with at least one field outside a threshold are reported.

The following topics provide additional information:

- "Output from exception reporting"
- "Specifying exceptions using the Exception Threshold data set editor" on page 59
- "How DB2 instrumentation data matches exception thresholds" on page 63
- "Example of producing an Accounting exception report" on page 64
- "Which exception fields and threshold values to choose" on page 66

Output from exception reporting

Exception processing provides several types of output as part of Accounting and Statistics reporting.

Exception reports, traces, and files are obtained using the EXCEPTION subcommand option in the ACCOUNTING and STATISTICS commands.

Reports and traces

Exception reports and traces are like the usual Accounting and Statistics reports and traces, except that they only contain records that have at least one field in exception status.

Logs

Exception logs combine both Accounting and Statistics fields that are in exception status into a single report and show the information in timestamp order.

File data sets and Log File data sets

The Exception File data set and the Exception Log File data set are sequential data sets that are suitable for use by the DB2 LOAD utility. The Exception File data set contains Accounting or Statistics records that have at least one field in exception status. The Exception Log File data set contains both Accounting and Statistics fields that are in exception status. The Exception File data set corresponds to an exception trace, whereas the Exception Log File data set corresponds to an exception log.

The Exception Log data set is written if you define the EXTRCDD1 DD statement in the JCL. To prevent the generation of the Exception Log data set, omit the EXTRCDD1 statement from your JCL (the preferred method), or specify DUMMY in the definition.
The Exception Log File data set is written if you define the EXFILDD1 DD statement in the JCL. To prevent the generation of the Exception Log File data set, omit the EXFILDD1 statement from your JCL (the preferred method), or specify DUMMY in the definition.

Specifying exceptions using the Exception Threshold data set editor

Exceptions are specified as threshold values in an Exception Threshold data set. When exception processing is active, DB2 instrumentation data is checked against these values and fields that contain values outside the specified thresholds are reported.

About this task

This topic describes how to specify threshold values with the Exception Threshold data set editor, which is used to set and maintain thresholds in an Exception Threshold data set.

The threshold values in the Exception Threshold data set can also be modified by means of the exception profiling method, which uses a sample of DB2 instrumentation data to calculate and set individual threshold values in the Exception Threshold data set. Refer to "Exception profiling" on page 178 for more details. For more details about the Exception Threshold data set and a sample data set that can be used to get started, refer to "Exception Threshold data set" on page 177.

The Exception Threshold data set editor is accessed indirectly from the Data Set Maintenance Menu (DGOPMENU, Figure 30). To display this menu, select option 4 (Maintain parameter data sets) from the Online Monitor Main Menu.

```
DGOPMENU Data Set Maintenance Menu

Select one of the following.

1  1. Maintain exception thresholds
   2. Maintain correlation translations
   3. Maintain time zone information
   4. Maintain MAINPACK definitions

Exception data set
'DGO710.THRESH' _______________________

DPMPARMS data set
'DGO710.DPMPARMS' ____________________
```

Figure 30. Selecting Maintain Exception Threshold data set

Type 1 in the input field to select Maintain exception thresholds, and type the name of your Exception Threshold data set on the line below Exception data set.

Press Enter. The Exception Threshold Category Selection panel is displayed, as shown in Figure 31 on page 60.
This panel shows the categories of exception threshold fields from which you can choose. The name of the category indicates the area where the exception applies:

- ... per Plan ...
- ... per Program ...
- ... per System ...
- ... per Address Space...

Exceptions are reported in Accounting or Statistics.

You can select any number of categories by typing a forward slash (/) or S in the selection field.

Fields that have previously been selected are marked by a greater than symbol (>). If you overtype the symbol in front of a category with a blank, the underlying selections are not deleted, but they are not used when exception reports are generated. To activate the category, select it again.

As an example, the Elapsed, CPU and Waiting Times per Plan Execution field is selected in Figure 31. When you press Enter, the Exception Threshold Field Selection panel is displayed, which shows all fields associated with this category, as shown in Figure 32 on page 61.
This panel shows all fields in this category. Fields that have been selected previously are marked by a greater than symbol (>) and sorted to the top.

Select a field for which you want to specify the exception thresholds. Type a forward slash (/) or $ in the selection field and press Enter. The Exception Threshold Field Details panel is displayed [Figure 33 on page 62].
Use the Exception Threshold Field Details panel to specify the threshold criteria in the Exception Threshold data set for the field selected from the Exception Threshold Field Selection panel.

The top right-hand side of this panel shows the entry number for this specification. You can specify more than one entry for the same exception field by using different criteria. For example, you can specify different exception thresholds for different plans. Or you might want to specify different threshold values for different environments (such as batch, online, or CICS). To add a new entry, you use the ADD command or F5 (Add). To view the different entries, use F10 (Previous) and F11 (Next). Remove entries that are no longer needed to avoid extra processing by using the DELETE command or F6 (Delete).

The first three fields in the panel show the category of the selected exception field, the field identifier, and the description of the field.

In this panel you can specify:
- Whether you want this field to be Active when exception reports are run. If you do not want to use the exception field specification when you generate exception reports the next time, but want to keep the entry, specify 2 (No).
- How you want the field value to be calculated before it is checked against the exception thresholds (By). If you want the value in the field used as is, without any calculation, specify Total. Alternatively, you can specify that the value for the field is divided by minutes, by seconds, or by the number of Commits, or you can specify that the exception threshold is checked for the average value per thread (this is especially useful in reports). The effect of these specifications varies according to the report or trace you produce.
- The Compare operator. It can be greater than or less than the threshold.
The thresholds. You can specify two kinds of thresholds for a field: Warning threshold and Problem threshold. Specify a warning threshold value to alert you to potential problems and a problem threshold value to indicate a more serious condition.

The additional criteria for the data for which the exception thresholds apply. These criteria are useful if you want to specify different exception threshold values depending on the environment. For example, you might want to specify different Elapsed Time thresholds for online transactions and batch jobs, in which case you would supply a specific connection name in this panel, press F5 (Add), and specify a different connection name in the second panel.

You can specify a generic name using an asterisk (\*).

Usage Notes:
- Specify the time value in seconds, for example, 90 to specify a threshold of 1 minute and 30 seconds.
- Do not enter 2 or 3 in the By field for time values. Specifying By Minute or By Second only is appropriate for fields that are not time related.
- For Accounting exception traces, By Thread has the same effect as Total because in an Accounting trace only one thread is used for one entry.
- For Accounting exception reports, if you specify Total, the threshold is checked for each thread. If you specify By Thread, the threshold value is checked against the average of the threads that are reported in one entry.
- The following applies to Statistics exception traces and reports:
  - If you specify By Minute or By Second, DB2 PE uses the value of the INTERVAL ELAPSED field in the HIGHLIGHTS block as a divisor to calculate the rate value that is to be checked against the defined threshold.
  - If you specify By Thread, the value of the THREADS field in the HIGHLIGHTS block is used as a divisor to calculate the rate value that is to be checked against the defined threshold.
  - If you specify By Commit, the value of the COMMITS field of the HIGHLIGHTS block is used as a divisor to calculate the rate value that is to be checked against the defined threshold.

How DB2 instrumentation data matches exception thresholds
During exception processing, DB2 instrumentation data records are tested against matching threshold conditions in the Exception Threshold data set.

When a record matches more than one threshold, the best matching threshold is chosen of those detecting an exception.

The key fields of a record are:
- Location
- Group
- Subsystem ID
- Member
- Requesting Location
- Connection
- Plan Name
- Correlation Name
- Correlation Number
- Primary Authorization ID
These key fields are compared with the corresponding fields in the thresholds and the threshold that best matches the record is chosen. This is done by comparing the Location of the record with the Location field of the matching thresholds first. A key field that does not contain an asterisk is considered a better match than one that contains asterisks. If two thresholds are found to match the record equally, the Group fields are compared with the record. If these fields also match the record equally, the Subsystem ID fields are compared with the record, and so on for the following fields:

- Member
- Requesting Location
- Connection
- Plan Name
- Correlation Name
- Correlation Number
- Primary Authorization ID

Exception checking in the reporting phase is performed on entries that are constructed according to your ORDER specification, which contains up to three OMEGAMON XE for DB2 PE identifiers. For example, if you qualify a field by PLANNNAME and CONNECT, exception checking on the field is performed only if you specified both PLANNNAME and CONNECT in ORDER.

Package-related fields are supported on a “by total” and “by thread” basis only.

**Example of producing an Accounting exception report**

The example shows the benefits of using exception reporting. It also describes how to produce an Accounting exception report.

Assume that you would regularly produce a short Accounting report, such as the example shown in Figure 34, and that the report would normally be fairly long.

If you want to use this Accounting exception report to find out whether the class 1 elapsed time per thread is unacceptably high, you would have to check every entry on every page even though you are not interested in entries that fall within normal bounds.

To save time, you can specify exception thresholds for the class 1 time field per thread, and run Accounting exception reports that show only entries that have exceptionally high values for this field. You can further limit the data by only checking online transactions because you would expect to find a high elapsed time for batch jobs.
In this example transactions have a plan name PLANTRNn and batch jobs have a plan name PLANBATn. The connection ID for the CICS address space is CICSA.

Access data set member RKO2DATA(DGOETV41) with the IRF and specify the thresholds in the Exception Threshold Field Details panel.

Figure 35. Specifying exceptions

This example specifies that:

- Exceptions are checked for the class 1 elapsed time field ADRECETT.
- The Elapsed Time thresholds are checked for the average value per thread. This means that the value in this field is divided by the number of threads.
- The comparison is Greater than.
- If the class 1 elapsed time value exceeds three seconds per thread, it is flagged as a warning exception. If it exceeds six seconds per thread, it is flagged as a problem exception.
- Only data with a connection ID of CICSA (CICS transactions) is checked for exceptions.

After you have completed the specifications, press Enter. The exception threshold specification is complete.

Now you can run the Accounting exception report. Remember to specify the name of your Exception Threshold data set for the EXCPTDD ddbname in the JCL. Use the following command to produce the report:

```
ACCOUNTING
   REPORT
```
This is what the Accounting exception report looks like:

![Accounting exception report](image)

The exception report is much shorter than the original Accounting report. It only lists transactions with class 1 elapsed times that exceeded the exception threshold, and it flags these transactions as warnings or problems.

If you specified more than one exception threshold, all records that contained any fields that reached an exception threshold value are listed.

**Which exception fields and threshold values to choose**

You can choose from a comprehensive set of exception fields and it can be difficult to decide which ones to choose and which threshold values to specify for your site. As a rule, most sites only need to define a limited number of thresholds.

To get started with exception reporting, consider using the sample Exception Threshold data set member RKO2DATA(DGOETV41), which contains a selection of predefined exception thresholds. For more information, see "Exception Threshold data set" on page 177.

In general, long response times are a good indicator of a performance problem and therefore you should start by defining exception thresholds for time fields.
To use exception processing efficiently, consider what the most important applications or transactions in your system are. Always define exception thresholds for critical business applications. In addition, frequently executed applications are good candidates for exception thresholds.

The application-specific thresholds are defined by specifying the plans for which the threshold applies. An efficient way of determining which plans or connection IDs should be the focus of exception reporting is to produce Accounting TOP lists.

You can use the performance objectives stated in your service level agreement as a starting point. Accounting TOP lists and TOP ONLY reports are good references when determining which threads to monitor with exception processing. You can modify the predefined threshold values and specify additional exception fields.

Carefully consider the fields for which to specify exception thresholds. The more fields you specify, the greater the effects on processing.

You can use the exception profiling method and a sample of your installation's DB2 instrumentation data to calculate threshold values for exception fields. For more information, see "Exception profiling" on page 178.

### Using SQL Activity reports to monitor SQL performance

SQL Activity reports provide detailed information about individual SQL statements in a plan or package. The SQL Activity report aggregates threads by a combination of OMEGAMON XE for DB2 PE identifiers. The SQL Activity trace shows details on a per thread basis.

At its simplest, the SQL Activity trace report shows each SQL statement in a thread by order of occurrence.

For each SQL statement basic information about the statement is shown. At the end of the thread, a summary is shown. If the thread originated at a different location, location information is also provided. A new thread is reported on a new page.

The end of the report shows a list of threads, in the order of occurrence, with a cross reference to the page where they are reported.

You can increase the level of detail by requesting *workload*. Workload shows information about DB2 activity associated with the execution of the statement. You can request workload details for:

- All workload
- Data capture
- Exits
- I/O activity
- Locking activity for rows and pages
- Scan activity
- Sort activity
- Host variables data

By default, no workload information is shown.

When you request workload, extra information is also provided:

#### Accounting

If IFCID 3 is included in your DB2 trace, OMEGAMON XE for DB2 PE includes an long Accounting trace for each thread.
Minibind
This shows bind information for each plan step. This is information taken from IFCID 22 and shows much of the information contained in the PLAN_TABLE.

User-defined functions
When user-defined functions are used, the report includes a Function Resolution block. This block shows information about the query, the path used, and detailed information about the function.

You can change the organization of data to group (summarize) events by:
• Cursor
• Program
• Statement number
• Statement type
or all of the above.

You can also sort the data by:
• Time:
  – Elapsed
  – Exit
  – I/O
  – Suspension
  – TCB
• Number of:
  – Exits
  – I/O requests
  – Merge passes for each sort
  – Pages scanned
  – Records sorted
  – Rows processed
  – Scans
  – Suspensions
  – Work files sorted

The following topics provide additional information:
• “When to use SQL Activity reports”
• “Input for SQL Activity reports” on page 69
• “Collecting SQL Activity data with a DB2 trace” on page 71
• “Creating an OMEGAMON XE for DB2 PE SQL Activity trace” on page 71
• “Interpreting the result” on page 74
• “Remark about lengths of SQL statement texts in SQL Activity traces” on page 75

When to use SQL Activity reports
Use SQL Activity reports to resolve performance problems with applications or queries that cannot be resolved by using the Accounting or Explain reports.

These are most commonly problems with scan and sort activity. To isolate the problem, you need to run an SQL Activity trace, which shows the workload for sort.
**Input for SQL Activity reports**

Each DB2 trace type and trace class contains one or more Instrumentation Facility Component identifiers (IFCIDs). The SQL Activity report set uses DB2 trace class and IFCID information to register specific SQL activity events.

SQL activity events are classified as follows:

- Accounting
- Base
- Data capture
- Exit
- I/O
- Locking
- Scan
- Sort
- Workload highlights
- Host variables data

An SQL activity event is normally characterized by a start record and an end record. When you select event records as input for your reports, you must include the paired end event, where applicable. When you use the ISPF monitor or workstation client Collect Report Data function to manage DB2 traces, the trace generator automatically includes the correct end record.

The following table shows the IFCIDs used in SQL Activity reporting, together with the associated DB2 trace classes and end pair IFCIDs, where appropriate.

*Table 7. IFCIDs used for SQL Activity reports*

<table>
<thead>
<tr>
<th>Start IFCID</th>
<th>DB2 trace class &amp; SQL Activity event type</th>
<th>End IFCID</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 Accounting</td>
<td>• 1 Accounting data</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• 2 DB2 times</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 3 Suspension, system events</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 5 Package in DB2 time</td>
<td></td>
</tr>
<tr>
<td>6 Read I/O</td>
<td>4 I/O</td>
<td>7 Read I/O end</td>
</tr>
<tr>
<td>8 Synch.</td>
<td>4 I/O</td>
<td>9 Synch. write end</td>
</tr>
<tr>
<td>write begin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Validation exit</td>
<td>13 Exit</td>
<td></td>
</tr>
<tr>
<td>12 Encode edit exit</td>
<td>13 Exit</td>
<td></td>
</tr>
<tr>
<td>15 Index scan begin</td>
<td>8 Scan</td>
<td>18 Common scan end</td>
</tr>
<tr>
<td>16 Insert scan begin</td>
<td>8 Scan</td>
<td>18 Common scan end</td>
</tr>
<tr>
<td>17 Sequential scan begin</td>
<td>8 Scan</td>
<td>18 Common scan end</td>
</tr>
<tr>
<td>19 Decode edit exit</td>
<td>13 Exit</td>
<td></td>
</tr>
<tr>
<td>20 Page and row locking summary</td>
<td>6 Locking</td>
<td>None</td>
</tr>
<tr>
<td>22 Minibind</td>
<td>3 Base</td>
<td>None</td>
</tr>
<tr>
<td>28 Sort runs</td>
<td>9 Sort</td>
<td>None</td>
</tr>
<tr>
<td>44 Lock suspend</td>
<td>6 Locking</td>
<td>45 Lock resume</td>
</tr>
<tr>
<td>53 SQL Describe, Commit, Rollback, or remote statement</td>
<td>3 Base</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 7. IFCIDs used for SQL Activity reports (continued)

<table>
<thead>
<tr>
<th>Start IFCID</th>
<th>DB2 trace class &amp; SQL Activity event type</th>
<th>End IFCID</th>
</tr>
</thead>
<tbody>
<tr>
<td>55 Set SQLID</td>
<td>3 Base</td>
<td>None</td>
</tr>
<tr>
<td>59 Fetch begin</td>
<td>3 Base</td>
<td>58 Common SQL end</td>
</tr>
<tr>
<td>60 Select begin</td>
<td>3 Base</td>
<td>58 Common SQL end</td>
</tr>
<tr>
<td>61 Insert, Update, or Delete Begin</td>
<td>3 Base</td>
<td>58 Common SQL end</td>
</tr>
<tr>
<td>63 SQL statement to be parsed</td>
<td>3 Base</td>
<td>None</td>
</tr>
<tr>
<td>62 DDL or DCL begin</td>
<td>3 Base</td>
<td>58 Common SQL end</td>
</tr>
<tr>
<td>64 Prepare begin</td>
<td>3 Base</td>
<td>58 Common SQL end</td>
</tr>
<tr>
<td>65 Open begin</td>
<td>3 Base</td>
<td>58 Common SQL end</td>
</tr>
<tr>
<td>66 Close begin</td>
<td>3 Base</td>
<td>58 Common SQL end</td>
</tr>
<tr>
<td>68 Rollback begin</td>
<td>3 Base</td>
<td>69 Rollback end</td>
</tr>
<tr>
<td>70 Commit phase 2 begin</td>
<td>3 Base</td>
<td>71 Commit phase 2 end</td>
</tr>
<tr>
<td>72 Create thread begin</td>
<td>3 Base</td>
<td>73 Create thread end</td>
</tr>
<tr>
<td>74 Terminate thread begin</td>
<td>3 Base</td>
<td>75 Terminate thread end</td>
</tr>
<tr>
<td>84 Commit phase 1 begin</td>
<td>3 Base</td>
<td>85 Commit phase 1 end</td>
</tr>
<tr>
<td>86 Signon begin</td>
<td>3 Base</td>
<td>87 Signon end</td>
</tr>
<tr>
<td>88 Synch. begin</td>
<td>3 Base</td>
<td>89 Synch. end</td>
</tr>
<tr>
<td>92 Access Method Services (AMS) begin</td>
<td>3 Workload highlight</td>
<td>97 AMS end</td>
</tr>
<tr>
<td>95 Sort begin</td>
<td>3 Sort</td>
<td>96 Sort end</td>
</tr>
<tr>
<td>105 DBID/OBID</td>
<td>10 Base</td>
<td></td>
</tr>
<tr>
<td>106 System parameters</td>
<td>8 Scan</td>
<td>18 Common scan end</td>
</tr>
<tr>
<td>107 Data set open or close</td>
<td>10 Base</td>
<td></td>
</tr>
<tr>
<td>108 Autobind begin</td>
<td>10 Base</td>
<td>109 Autobind end</td>
</tr>
<tr>
<td>125 RID list processing</td>
<td>8 Scan</td>
<td>18 Common scan end</td>
</tr>
<tr>
<td>108 Autobind begin</td>
<td>10 Base</td>
<td>109 Autobind end</td>
</tr>
<tr>
<td>157 DRDS RDS interface</td>
<td>16 Base</td>
<td>None</td>
</tr>
<tr>
<td>159 DRDS CNV interface</td>
<td>16 Base</td>
<td>None</td>
</tr>
<tr>
<td>160 DC requester</td>
<td>16 Base</td>
<td>None</td>
</tr>
<tr>
<td>161 DC server</td>
<td>16 Base</td>
<td>None</td>
</tr>
<tr>
<td>162 DTM request</td>
<td>16 Base</td>
<td>None</td>
</tr>
<tr>
<td>163 DTM response</td>
<td>16 Base</td>
<td>None</td>
</tr>
<tr>
<td>174 Archive log (Quiesce) suspend</td>
<td>3 Base</td>
<td>175 Archive log (Quiesce) resume</td>
</tr>
<tr>
<td>177 Package allocation</td>
<td>3 Base</td>
<td>None</td>
</tr>
<tr>
<td>183 DRDS RDS/SCC interface invocation</td>
<td>16 Base</td>
<td>183 DRDS RDS/SCC interface return</td>
</tr>
<tr>
<td>185 READs data capture start</td>
<td>16 Base</td>
<td>188 READs data capture end</td>
</tr>
<tr>
<td>213 Drain lock wait begin</td>
<td>6 Locking</td>
<td>214 Drain lock wait end</td>
</tr>
</tbody>
</table>
Table 7. IFCIDs used for SQL Activity reports (continued)

<table>
<thead>
<tr>
<th>Start IFCID</th>
<th>DB2 trace class &amp; SQL Activity event type</th>
<th>End IFCID</th>
</tr>
</thead>
<tbody>
<tr>
<td>215</td>
<td>Claim count zero wait begin</td>
<td>216</td>
</tr>
<tr>
<td>218</td>
<td>Lock avoidance summary</td>
<td>None</td>
</tr>
<tr>
<td>221</td>
<td>Parallel group degree</td>
<td>18</td>
</tr>
<tr>
<td>222</td>
<td>Parallel group elapsed</td>
<td>18</td>
</tr>
<tr>
<td>226</td>
<td>Page latch wait begin</td>
<td>227</td>
</tr>
<tr>
<td>231</td>
<td>Parallel group task time</td>
<td>18</td>
</tr>
<tr>
<td>233</td>
<td>Call Stored Procedure begin</td>
<td>58</td>
</tr>
<tr>
<td>237</td>
<td>Set current degree</td>
<td>58</td>
</tr>
<tr>
<td>239</td>
<td>Overflow package/DBRM</td>
<td>None</td>
</tr>
<tr>
<td>247</td>
<td>Input Host Variable</td>
<td>None</td>
</tr>
<tr>
<td>272</td>
<td>Associated Locators begin</td>
<td>58</td>
</tr>
<tr>
<td>273</td>
<td>Allocate Cursor begin</td>
<td>58</td>
</tr>
<tr>
<td>305</td>
<td>Table check constraint</td>
<td>18</td>
</tr>
<tr>
<td>324</td>
<td>Function resolution (UDF)</td>
<td>58</td>
</tr>
<tr>
<td>325</td>
<td>Trigger</td>
<td>58</td>
</tr>
</tbody>
</table>

Collecting SQL Activity data with a DB2 trace

To create an SQL Activity trace report, you need to collect data by using a DB2 trace.

The easiest way to collect data for any OMEGAMON XE for DB2 PE report is to use the activate traces facility of the workstation monitor. You can use this facility to configure a DB2 trace and start and stop the trace from there. Using the activate traces facility also ensures that you include all data in the DB2 trace that is needed for a report. For more information, see Monitoring Performance from Performance Expert Client.

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE (P) CLASS (30) RMID (*) DEST (OPX) PLAN (PMOMDEV) AUTHID (DB2PE)
 IFCID(6,7,8,9,11,12,13,15,16,17,18,22,28,45,53,55,58,59,60,
  61,62,63,64,65,66,68,69,70,71,72,73,74,75,78,80,85,88,87,88,89,92,95,96,97,
```

Because you need sort data, include IFCIDs 95 and 96 (Sort Start and Sort End).

For information regarding DB2 trace data provided by IFCID 350 see "Remark about lengths of SQL statement texts in SQL Activity traces" on page 75.

Creating an OMEGAMON XE for DB2 PE SQL Activity trace

Use your ISPF editor to create a JCL job that creates an SQL Activity trace.
The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.

```
EDIT ---- ANDREW.OMPE.JOBS.SQLTRC--------------------- Columns 001 072
Command >>> Scroll ===> CSR_
****** ******************************* TOP OF DATA *******************************
0001 /* JCL to produce an SQL activity trace */
0002 //ANOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=Y,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //SYSIN DD *
0007 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE2,DISP=SHR
0008 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.SQLT01,DISP=OVR
0009 //JOBSUMDD DD SYSOUT=A
0010 SQLACTIVITY
0011 TRACE
0012 SUMMARIZEBY (OCCURRENCE)
0013 WORKLOAD (SORT)
0014 EXEC
****** ******************************* BOTTOM OF DATA ****************************
```

Figure 37. SQL Activity job stream using ISPF/PDF editor

To submit the job, type SUBMIT on the command line and press Enter.

Here is an example of an SQL Activity trace, summarized by occurrence.
<table>
<thead>
<tr>
<th>Event</th>
<th>Timestamp</th>
<th>Elapsed Time</th>
<th>TCB Time</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBRM</td>
<td>PYPIA</td>
<td>20:16:13.13</td>
<td>0.000162</td>
<td>STMT# 408 ISO(CS) SQLSTATE: 00000 SQLCODE: 0 REOPTIMIZED(NO) KEEP UPDATE LOCKS: N/A</td>
</tr>
<tr>
<td>OPEN</td>
<td></td>
<td>20:16:13.13</td>
<td>0.000166</td>
<td>STMT# 416 CURSOR: CUR_CUSTNAME ISO(CS) SQLSTATE: 00000 SQLCODE: 0 REOPTIMIZED(NO) KEEP UPDATE LOCKS: NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WORKLOAD HILITE -----------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SCANS : 2 RECS/SORT: 1.00 I/O REQ'S: 1 SUSPENDS : N/P EXITS : N/P AMS : N/P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ROWSPROC: 2 WORK/SORT: 1.00 AET/1/O : 0.000005 AET/SUSP : N/P AET/EXIT : N/P AET/AMS : N/P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PAGESCAN: 30 PASS/SORT: 0.00 DATACAPT: N/P RID'S UNUSED: N/P CHECKCON : N/P DEGREE REDUCTION : N/P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LOB_PAGSCAN: 0 LOB_UPD_PAGE : 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SORT TYPE : ESA PARTITION TYPE : NONE MERGE PASSES: 0.00</td>
</tr>
<tr>
<td>FETCH</td>
<td></td>
<td>20:16:13.13</td>
<td>0.000065</td>
<td>STMT# 425 CURSOR: CUR_CUSTNAME ISO(CS) SQLSTATE: 00000 SQLCODE: 0</td>
</tr>
<tr>
<td>UPDATE</td>
<td></td>
<td>20:16:13.15</td>
<td>0.000130</td>
<td>STMT# 483 CURSOR: CUR_CUSTOMER ISO(CS) SQLSTATE: 00000 SQLCODE: 0 REOPTIMIZED(NO) KEEP UPDATE LOCKS: N/A</td>
</tr>
<tr>
<td>OPEN</td>
<td></td>
<td>20:16:13.15</td>
<td>0.00015</td>
<td>STMT# 493 CURSOR: CUR_WAREHOUSE ISO(CS) SQLSTATE: 00000 SQLCODE: 0 REOPTIMIZED(NO) KEEP UPDATE LOCKS: NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WORKLOAD HILITE -----------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SCANS : 2 RECS/SORT: 1.00 I/O REQ'S: 1 SUSPENDS : N/P EXITS : N/P AMS : N/P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ROWSPROC: 2 WORK/SORT: 1.00 AET/1/O : 0.000006 AET/SUSP : N/P AET/EXIT : N/P AET/AMS : N/P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PAGESCAN: 30 PASS/SORT: 0.00 DATACAPT: N/P RID'S UNUSED: N/P CHECKCON : N/P DEGREE REDUCTION : N/P</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LOB_PAGSCAN: 0 LOB_UPD_PAGE : 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SORT TYPE : ESA PARTITION TYPE : NONE MERGE PASSES: 0.00</td>
</tr>
<tr>
<td>FETCH</td>
<td></td>
<td>20:16:13.15</td>
<td>0.000094</td>
<td>STMT# 497 CURSOR: CUR_WAREHOUSE ISO(CS) SQLSTATE: 00000 SQLCODE: 0</td>
</tr>
<tr>
<td>UPDATE</td>
<td></td>
<td>20:16:13.15</td>
<td>0.000071</td>
<td>STMT# 507 CURSOR: CUR_WAREHOUSE ISO(CS) SQLSTATE: 00000 SQLCODE: 0 REOPTIMIZED(NO) KEEP UPDATE LOCKS: N/A</td>
</tr>
<tr>
<td>INSERT</td>
<td></td>
<td>20:16:13.15</td>
<td>0.000096</td>
<td>STMT# 544 ISO(CS) SQLSTATE: 00000 SQLCODE: 0 REOPTIMIZED(NO) KEEP UPDATE LOCKS: N/A</td>
</tr>
</tbody>
</table>

Figure 38. SQL Activity trace, summarized by occurrence (part 1)
Interpreting the result

The example shows details of the Sort Activity block of an SQL Activity trace.

Following the example in this section, you suspect that the application has problems with sorting or scanning data. The Sort Activity block of the report is printed under the event that triggered the sort, together with the workload highlights, and any other requested workload.

Figure 41. SQL Activity workload highlight block

Verify the average elapsed times for sorts (AET/SORT field). Approximately 0.00001 seconds (0.01 milliseconds) per row sorted is acceptable. Significantly higher values indicate a sort problem.
Sort problems could be caused by any of the following factors:
- Statements with DISTINCT
- Noncorrelated subqueries
- UNION
- Missing index

If your analysis indicates no significant sort problems, the problem could be a scan problem. Look for SQL statements with:
- Nonindexable predicates (WHERE (COL1*COL2) > (COL3*COL4))
- OR connections
- Noncorrelated subqueries
- Use of static SQL rather than dynamic SQL
- Missing index

**Remark about lengths of SQL statement texts in SQL Activity traces**

How extended SQL statement texts become visible in SQL Activity traces with IFCID 350.

An SQL Activity trace might show SQL statement texts in the report, depending on the subcommand options that were specified with the SQLACTIVITY TRACE command. The SQL statement texts are provided by IFCID 63, which limits the length to a maximum of 5000 characters. Consequently, SQL statement texts shown in SQL Activity trace reports are also limited to this length.

With DB2 version 8, IFCID 350 was introduced. This IFCID records SQL statement texts up to a length of 200000 characters each. However, IFCID 350 data will not be shown in SQL Activity reports and traces, independently of whether IFCID 350 data is provided by a DB2 trace.

To obtain extended SQL statement texts, perform the following steps:
1. Start a DB2 trace to collect data, using trace class 30, 31, or 32, and specifying IFCID 350, for example:
   ```
   -START TRACE CLASS(30) IFCID(350) DEST(SMF) TDATA(CPU,COR,DIST)
   ```
2. Create a Record Trace report (refer to "Record Trace reports" on page 101 for details).
   Use the GLOBAL command and its options to minimize the amount of data that is produced by the Record trace and to filter the required data, similar to the following example:
   ```
   GLOBAL
   FROM(,12:19:59.99)
   TO(,12:20:00.93)
   INCLUDE(IFCID(350))
   ```

**Using Explain reports to monitor access paths**

The OMEGAMON XE for DB2 PE Explain reports are built on the information that is supplied by the DB2 SQL Explain function and information taken from the system catalog tables. An Explain report provides information about the exact access path that is used by DB2 for any particular SQL statement and its effect on DB2 performance.

Because DB2 is a relational database system, masses of data are stored in the form of table columns with no predetermined row order. Data can also be spread across several tables, partitions, and DB2 instances. The arrangement of data, and the complexity of many queries means that the method (access plan) used, and the
The route taken to the desired data (access path) cannot always be determined at application development time, especially when dynamic SQL is used.

The access plan and access path used by DB2 are important factors in DB2 performance. However, for any particular query, the access plan and access path are influenced by many factors, such as whether indexes are used and, if so, how many, whether data is clustered, joins are used, the locking strategy used, and so on.

The following topics provide additional information:

- “Explain reports”
- “Levels of detail” on page 77
- “Required authorization” on page 78
- “When to use Explain reports” on page 78
- “Creating an OMEGAMON XE for DB2 PE Explain report” on page 78
- “Explain report example” on page 79
- “Interpreting the Explain report” on page 85

**Explain reports**

OMEGAMON XE for DB2 PE provides Explain reports by query number, statement text, plan, package, and QMF query.

**Query number**

You can investigate a particular SQL statement identified by its query number when:

- A dynamic SQL EXPLAIN statement was executed with a given query number. The statement can be from DB2 or QMF. If the query number is not specified, DB2 assigns a number. You can get the query number from the PLAN_TABLE.
- The application was bound (or rebound) with EXPLAIN(YES). In this instance, the statement query number is assigned by the precompiler.

Explain searches for the query number in the job submitter’s PLAN_TABLE, unless a different owner is specified. If you want to run an Explain report for a statement that you do not own, you need access to the owner’s plan table.

**Statement text**

Can be used to investigate a particular SQL statement that is supplied in its text form.

Explain uses 999 735 911 as a query number. If this statement number already exists in the PLAN_TABLE, the row is deleted before processing the SQL statement. After successful execution of SQL EXPLAIN, the newly inserted row in the PLAN_TABLE is used to produce the Explain report.

**Plan**

This reports on all or selected SQL statements contained in an application plan.

To create a report for a plan, that plan must exist in the catalog table SYSIBM.SYSPLAN and must have been bound with EXPLAIN(YES).

**Package**

This reports on SQL statements in a package.
To create a report for a package, that package must exist in the catalog table SYSIBM.SYSPACKAGE and must have been bound with EXPLAIN(YES).

QMF query

Can be used to investigate a saved QMF query written in SQL. QBE and PROMPTED queries must be converted to SQL before they can be explained.

You can create reports about your own saved QMF queries, and queries created by other users, saved with SHARE=YES.

The QMF query can contain parameters, for example, and &PARM1 and &PARM2. These parameters can also substitute column names in the select list. However, the query must not contain literals and other strings with one or more ampersands (&), enclosed between quotes (').

If there is more than one entry in the PLAN_TABLE with the same identifiers, the most recent entry is reported.

Levels of detail

To control the amount of data, which can be very large especially for explain plan or package, you can request summary and detail level Explain reports.

Summary

A summary report shows one line for each SQL statement.

A summary block is always printed at the end of a report.

Detail

This produces a full report for each SQL statement, which shows:

- “Raw” SQL EXPLAIN data as found in the PLAN_TABLE
- Access path data
- Table and table space data
- Index data, this shows:
  - All available indexes for a given table
  - Indexes selected for the access only
- Key data
- Plan and package data
- Host variables data
- Summary page

The number of data blocks listed varies with the Explain function requested.

Basic

This shows raw data from the PLAN_TABLE and Access Path Data blocks in the report for each SQL statement. Information from the catalog tables is not included.

SQL

The Access Path Data block is shown for each SQL statement.

Index Data

All data blocks of the DETAIL level are shown except for the Key Data block.

No Raw PLAN_TABLE Data

All data blocks of the DETAIL level, except the raw PLAN_TABLE, are shown.

Key Distribution

All data blocks of the DETAIL level are shown including the distribution of the ten mostly used key values.
Required authorization

Explain reports require SELECT authorization to access certain tables.

Because OMEGAMON XE for DB2 PE uses DB2 system catalog tables to create Explain reports, you need SELECT authorization for the SYSIBM.SYS* catalog tables.

To explain saved QMF queries, you need SELECT authorization for the following QMF tables:
- Q.OBJECT_DIRECTORY
- Q.OBJECT_DATA

To control user access to the catalog tables, OMEGAMON XE for DB2 PE selects data through views. Sample definitions are in the RKO2SAMP library. You can use these definitions as supplied, or tailor them for your installation. For more information, see Configuration and Customization and the Program Directory.

When you produce an Explain report, OMEGAMON XE for DB2 PE checks for the plan table. If it does not exist, it is created in the default database, if you have the required privileges. If you are not authorized, the execution of Explain ends with an error message.

When to use Explain reports

Use Explain reports if you suspect a problem with indexes or SQL statements.

You should be able to identify candidate plans or packages from the Accounting report. Generally, an SQL statement or index problem is indicated when an application seems to be spending a lot of time in DB2. That is, when the class 2 elapsed time is not significantly greater (50% or more) than the class 2 CPU (TCB).

Access path problems can also be indicated by a high number of Getpage operations for an SQL statement.

If you are using dynamic SQL, you can use the EXPLAIN statement to obtain information about each statement.

For static SQL, unless you know the text of a suspect statement or the statement number, use the Explain reports at the plan or package level.

Creating an OMEGAMON XE for DB2 PE Explain report

Use your ISPF editor to create a JCL job that creates an Explain report.

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.
This example produces an Explain detail report for the second generation of a package (DGO@FIST) on DB2 subsystem DSNT.

### Explain report example

This example is an Explain report generated from the JCL in the previous section.

This example report has been split into its constituent report blocks for clarity. Similarly, the page header is only shown once for the report, and once for the report summary.

---

### Figure 42. Explain JCL

```plaintext
EDIT ---- ANDREW.OMPE.JOBS.EXPL1 "-------- Begin of columns 001 072
Command ==> Scroll ==> CSR_"

******************************************************************************
0001 /* JCL to produce an explain report */
0002 // AWOMPEA JOB (TTS1,YUS7), 'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 // STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 // SYSDT DD SYSDT=*
0007 // EXPLAIN DD SYSDT=*
0008 // DPMOUTDD DD DSN=ANDREW.OMPE.OUT.EXPL1,DISP=OVR
0009 // JOBSUMDD DD SYSDT=A
0010 EXPLAIN
0011 PACKAGE (PMDEVX.DGO@FIST.(-2))
0012 SSID(DSNT)
0013 HOSTVAR(YES)
0014 EXEC

******************************************************************************
```

---

### Figure 43. Explain report – page header

- **Actual AT:** 10/13/08 13:10:38
- **Location:** PMODB2A
- **Subsystem:** SDA2
- **User AuthID:** XRK
- **DB2 Version:** V8
- **Explain Plan:** Detail
- **Current SQLID:** XRK
FPEY0166I PACKAGE MYTEST IN COLLECTION COLLECT HAS THE FOLLOWING VERSIONS

PRE-COMP'D EXP GEN VERSION IDENTIFICATION
---------- --- ---- -----------------------
2002-08-24 YES 0 VERSION0
2002-08-23 YES -01 VERSION1
2002-08-22 YES -02 VERSION2
2002-08-21 YES -03 VERSION3
2002-08-20 YES -04 VERSION4
2002-08-19 YES -05 VERSION5
2002-08-18 YES -06 VERSION6
2002-08-17 YES -07 VERSION7
2002-08-16 YES -08 VERSION8
2002-08-15 YES -09 VERSION9
2002-08-14 YES -10 VERSION ID NOT SPECIFIED
2002-08-13 YES -11 VERSION11
2002-08-12 YES -12 VERSION12
2002-08-11 YES -13 VERSION13
2002-08-10 YES -14 VERSION14
2002-08-09 YES -15 VERSION15
2002-08-08 YES -16 VERSION16
2002-08-07 YES -17 VERSION17
2002-08-06 YES -18 VERSION18
2002-08-05 YES -19 VERSION19
2002-08-04 YES -20 VERSION20

START VERSION GENERATION NUMBER SPECIFIED: -7
NUMBER OF VERSION GENERATIONS REQUESTED: 3

Figure 44. Explain report – package version details

PACKAGE LOCATION :PM010171
PACKAGE COLLECTION:PMDEV
PACKAGE ID :DG0FIST
PACKAGE VERSION ID:R710_LEVEL2
STATEMENT NUMBER : 7
SQL STATEMENT READ FROM SYSIBM.SYSPACKSTMT:

```sql
SELECT VERSION INTO :DBRMVERSID
FROM DGO_SYSDBRM
WHERE NAME = :PROGNAME AND PLNAME = :HV_PLNAME
```

STATUS : COMPILED-BOUND USING DEFAULTS FOR INPUT VARIABLES
ISOLATION: UNCOMMITTED READ

---

Figure 45. Explain report – statement details

PLAN TABLE DATA  OWNER: PMDEV
BIND TIME : 2008-09-05-11.41.47.543347
QUERYNO: 7, ACC. TYPE: I, PLAN NO: 1, TABLE NAME: SYSDBRM
Q BLOCK NO.: 1, MATCHCOLS: 0, TAB. NO: 1, - OWNER: SYSIBM
DATE: 2008-09-05, PLAN: , METHOD : 0, INDEX NAME: PMSYSRDBRM
TIME: 11:41:47.5, PKGS: DG0FIST, IDXONLY : NO, - OWNER: DB3704
TS LOCKMODE: N, COL.FUNC. : , PREFETCH: , MULT. INDEX : 
ACCESS DEG : 0, ACC.PGROUP: 0, JOIN DEG: 0, JOIN PGROUP: 0
PAGE RANGE : , PARALL.MODE: , JOIN TYPE: , MERGE JOIN : 0
WHEN OPTIM : , PRIMARY ACCESSTYPE: , QBLOCK TYPE: SELECT
HINT_USED : , OPT_HINT_ID:
CORRELATION: , GROUP MEMB.:
COLLECTION : PMDEV , VERSION : R710_LEVEL2
SORTN TABLE -UNIQUE: N, JOIN: N, ORDER BY: N, GROUP BY: N, PARAL.GROUP: 0
SORTC TABLE -UNIQUE: N, JOIN: N, ORDER BY: N, GROUP BY: N, PARAL.GROUP: 0

---

Figure 46. Explain report – PLAN_TABLE data
THE ACCESS PATH CHOSEN BY DB2 AT 11:41:47.5 ON 2008-09-05

------------------------------------------------------------------
! NON-MATCHING INDEX SCAN WITH SCAN OF REFERENCED DATA PAGES !
! NON-CLUSTERED INDEX SCAN WILL BE USED !
! PAGE RANGE SCAN WILL NOT BE USED !
!
------------------------------------------------------------------

Figure 47. Explain report – access path

INDEX: DB3704.PMSYSRDBRM

<table>
<thead>
<tr>
<th>INDEX STAT</th>
<th>TYPE</th>
<th>Created</th>
<th>Altered</th>
<th>FRKCARD</th>
<th>PAGES</th>
<th>LEVELS</th>
<th>CLUSTERING</th>
</tr>
</thead>
<tbody>
<tr>
<td>7, PAGES: 1, LEVELS: 2, CLUSTERING: N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 48. Explain report – index details

KEY

<table>
<thead>
<tr>
<th>NO.</th>
<th>COLUMN NAME</th>
<th>COL.TYPE</th>
<th>LENGTH</th>
<th>NULL</th>
<th>CARD. ORDER</th>
<th>LOW2KEY</th>
<th>HIGH2KEY</th>
<th>USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PLCREATOR</td>
<td>CHAR</td>
<td>8</td>
<td>NO</td>
<td>ASC.</td>
<td>C'SYSADM</td>
<td>C'SYSADM</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>PLNAME</td>
<td>CHAR</td>
<td>8</td>
<td>NO</td>
<td>-1 ASC.</td>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 49. Explain report – key column details

TABLE: SYSIBM.SYSDBRM

<table>
<thead>
<tr>
<th>TABLE STAT</th>
<th>TYPE</th>
<th>ROWS</th>
<th>COLUMNS</th>
<th>ROWLENGTH</th>
<th>EDIT PROC.</th>
<th>ACT. PAGES</th>
<th>TABLE ID</th>
<th>STATUS</th>
<th>TABCREATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPL</td>
<td>TABL</td>
<td>7</td>
<td>16</td>
<td>193</td>
<td>NONE</td>
<td>7</td>
<td>35</td>
<td></td>
<td>SYSIBM</td>
</tr>
</tbody>
</table>

Figure 50. Explain report – table details

TABLESPACE: DSNDB06.SYSPLAN

<table>
<thead>
<tr>
<th>TABLESPACE</th>
<th>STATSTYPE</th>
<th>Created</th>
<th>Altered</th>
<th>PAGES ACTIVE</th>
<th>PSID</th>
<th>ERASERULE</th>
<th>DATABASE</th>
<th>PAGES SIZE</th>
<th>LOCKRULE</th>
<th>CLOSERULE</th>
<th>STOGROUP</th>
<th>PARTITIONS</th>
<th>LOCKPART</th>
<th>TS STATUS</th>
<th>BUFPOOL</th>
<th>SPACE</th>
<th>SEG.SIZE</th>
<th>TABLES/TS</th>
<th>TS TYPE</th>
<th>DEFAULT ENC.SCHEME</th>
<th>MAXROWS</th>
<th>LOG</th>
<th>DEFAULT SBCS CCSID</th>
<th>DEFAULT DBCS CCSID</th>
<th>MIX. CCSID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2008-08-05-17.22.36.236833</td>
<td>2008-08-05-17.22.36.236833</td>
<td>720, PSID: 10</td>
<td>5</td>
<td>NO</td>
<td>DSNDB06</td>
<td>4096</td>
<td>ANY, CLOSERULE: NO</td>
<td>NO, STOGROUP:</td>
<td>UNPART, LOCKPART: NO, TS STATUS: A, BUF_POOL: BPO</td>
<td>5, TS TYPE: NORMAL</td>
<td>127, LOG: YES</td>
<td>0</td>
<td>0</td>
<td>EBCDIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 51. Explain report – table space details

HOST VAR. TYPE LENGTH IND. HOST VARIABLE NAME

| VAR. CHARACTER | 64 NO | DBRMVERSID |
| FIXED CHARACTER | 8 NO | PROGNAME |
| FIXED CHARACTER | 8 NO | HV_PLNAME |

Figure 52. Explain report – host variables details
LOCATION : PMO1D711
COLLECTION ID: PMDEVX
PACKAGE ID : DGO@FIST
VERSION ID : R710_LEVEL2
CONSIST.TOKEN: X'16EBAA6A016D7A88'
PACKAGE NAME : D01M710.LEVEL2.SDGODBRM
OWNER : PMDEV QUOTE : APOSTROPHE
CREATOR : XRK COMMA : PERIOD
BIND DATE : 2008-09-05 HOSTLANG : ASSEMBLER
BIND TIME : 11.41.47.533590 CHARSET : ALPHANUMERIC
CREATE DATE : 2008-07-30 MIXED : NO
CREATE TIME : 10.20.31.320795 DEC31 : NO
QUALIFIER : PMDEV DATA CURRENCY: INHIBIT BLOCKING
BASE SIZE : 1672 SQLERROR : NOPACKAGE
AVERAGE SIZE : 4008 SOURCE : DBRM
SYSENTRIES : 0 PRECOMP. DATE: 2008-03-22
SQL STATEMENT: 1 PRECOMP. TIME: 12.55.21.867754
VALIDATE : BIND VALID : YES
ISOLATION : UNCOMMITTED READ OPERATIVE : YES
RELEASE : CHECK PLAN REOPTIMIZAT. : NO
DEGREE : 1 DEFERPREPARE : INHERITED FROM PLAN
KEEP DYNAMIC : DRDA DDF PROTOCOL : INHERITED FROM PLAN
TYPE OF PACK.: BIND PACKAGE OPT_HINT_ID : FNCT.RESOLVED: 2008-09-05-11.41.47.533590
.....5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0....5.. PATH:

Figure 53. Explain report – language and compile details

DECLARE C_SYSPACK_STMT CURSOR FOR SELECT SEQNO, STMT, ISOLATION, STATUS FROM DGO_SYSPACKSTM WHERE LOCATION = ' ' AND COLLID = :HV_IN_COLL AND NAME = :#0002 AND STMTNO = :#0001 AND VERSION = :HV_PACKVERSID ORDER BY SEQNO

STATUS : COMPILED-BOUND USING DEFAULTS FOR INPUT VARIABLES
ISOLATION: UNCOMMITTED READ

Figure 54. Explain report – statement details

PLAN TABLE DATA OWNER: PMDEV
BIND TIME : 2008-09-05-11.41.47.53347
QUERY NO: 259, ACC. TYPE: I, PLAN NO: 1, TABLE NAME: SYSPACKSTM
Q BLOCK NO.: 1, MATCHCOLS: 3, TAB. NO: 1, - OWNER: SYSIBM
DATE: 2008-09-05, PLAN: , METHOD: 0, INDEX NAME: DSNSXX01
TIME: 11:41:47.5, PKCG: DGO@FIST, IDONLY: NO, - OWNER: SYSIBM
TS LOCKMODE: N, COL.FUNC.: , PREFETCH: L, MULT.INDEX :
ACCESS DEG: 0, ACC.GROUP: 0, JOIN DEG: 0, JOIN GGROUP: 0
PAGE RANGE : , PARALL.MODE: , JOIN TYPE: , MERGE JOIN : 0
WHEN_OPTIM : , PRIMARY_ACCESSTYPE: , QBLCK.TYPE: SELECT
HINT_USED : , OPT_HINT_ID:
CORRELATION: , GROUP MEMB.: COLLECTION : PMDEVX , VERSION : R710_LEVEL2
SORTN TABLE : UNIQUE: N, JOIN: N, ORDER BY: N, GROUP BY: N, PARAL.GROUP: 0
SORTC TABLE : UNIQUE: N, JOIN: N, ORDER BY: N, GROUP BY: N, PARAL.GROUP: 0

Figure 55. Explain report – PLAN_TABLE details
THE ACCESS PATH CHOSEN BY DB2 AT 11:41:47.5 ON 2008-09-05

--- MATCHING INDEX SCAN WITH SCAN OF REFERENCED DATA PAGES ---
--- NUMBER OF MATCHING COLUMNS: 3 - THE INDEX HAS 5 COLUMNS ---
--- CLUSTERED INDEX SCAN WILL BE USED ---
--- PREFETCH TABLE SPACE SCAN THROUGH A PAGE LIST WILL BE PERFORMED ---
--- PAGE RANGE SCAN WILL NOT BE USED ---

Figure 56. Explain report – access path

INDEX: SYSIBM.DSNKSX01 --------------------------------------------------------
STATSTIME: 2008-08-05-17.22.22.515561
CREATED : 0001-01-01-00.00.00.000000 ALTERED: 0001-01-01-00.00.00.000000
FULL KEY CARD: 50453,PAGES : 1366,LEVELS: 4,CLUSTERING: Y
1ST KEY CARD: 1,SPACE : 18.678M,UNIQUE: YES,CLUSTERED: Y
INDEX TYPE : 2,PGSIZE: 4096,BFPOOL: BP0,DB.NAME : DSNDB06
CLUSTERRATIO : 97.9466%,ERRULE: NO,CLRULE: NO,IXSPACE : DSNKSX01
MAX.PIECESIZE: 0,COPY : NO,COPYLRSN: X'000000000000'

Figure 57. Explain report – index details

Table: SYSIBM.SYSPACKSTMT -----------------------------------------------------
STATSTIME: 2008-08-05-17.22.22.515561 TB TYPE : TABL
CREATED : 2008-04-01-00.00.00.000000 ALTERED : 2008-11-29-15.50.21.662242
ACTUAL AT:09/05/08 11:45:25 DB2PM (V7) PAGE : 1-7
EXPLAIN PACKAGE DB2 RELEASE: V7

Figure 58. Explain report – key column details

TABLESPACE: DSNDB06.SYSPKAGE --------------------------------------------------
STATSTIME: 2008-08-05-17.22.22.515561
CREATED : 0001-01-01-00.00.00.000000 ALTERED: 0001-01-01-00.00.00.000000
PAGES ACTIVE: 4320, PGSID: 121, ERASURE: NO, DATABASE : DSNDB06
PAGE SIZE : X, LOCKRULE: ANY, CLOSERULE: NO, STOGROUP :
PARTITIONS: UNPART, LOCKPART: NO, TS STATUS: A, BUF.POOL : BPO
SPACE : N/A, SEG.SIZE: 4, TABLES/TS: 8, TS TYPE : NORMAL
DEFAULT ENC.SCHEME: EBCDIC , MAXROWS : 255, LOG : YES
DEFAULT SBCS CCSID: 0, DEFAULT DBCS CCSID: 0, MIX. CCSID: 0

Figure 59. Explain report – table space details

Chapter 3. Using OMEGAMON XE for DB2 PE reports  83
### Figure 61. Explain report – host variables details

**HOST VAR. TYPE** | **LENGTH** | **IND.** | **HOST VARIABLE NAME**
--- | --- | --- | ---
FIXED CHARACTER | 18 | NO | HV_IN_COLL
FIXED CHARACTER | 8 | NO | #0002
LARGE INTEGER | 4 | NO | #0001
VAR. CHARACTER | 64 | NO | HV_PACKVERSID

### Figure 62. Explain report – statement details (repeated for second statement step)

```
DECLARE C_SYSPACK_STMT CURSOR FOR
SELECT SEQNO, STMT, ISOLATION, STATUS
FROM DGO_SYSPACKSTMT
WHERE LOCATION = ' ' AND COLLID = :HV_IN_COLL AND NAME =
#:0002 AND SMTNO = :#0001 AND VERSION = :HV_PACKVERSID
ORDER BY SEQNO
```

**STATUS**: COMPILED-BOUND USING DEFAULTS FOR INPUT VARIABLES
**ISOLATION**: UNCOMMITTED READ

### Figure 63. Explain report – PLAN_TABLE (for second statement step)

```
PLAN TABLE DATA OWNER: PMDEV
BIND TIME: 2008-09-05-11.41.47.543347
QUERYNO: 259, ACC. TYPE: , PLAN NO: 2, TABLE NAME:
Q BLOCK NO.: 1, MATCHCOLS: 0, TAB. NO: 0, - OWNER:
DATE: 2002-09-05, PLAN: , METHOD : 3, INDEX NAME:
TIME: 11:41:47.5, PKG: DGOOFIST, INDEXONLY : NO, - OWNER:
TS LOCKMODE: , COL.FUNC. : , PREFETCH: , MULTI_INDEX :
ACCESS DEG : 0, ACC.PGROUP: 0, JOIN DEG: 0, JOIN PGROUP: 0
PAGE RANGE : , PARALL.MODE: , JOIN TYPE: , MERGE JOIN : 0
WHEN_OPTIM : , PRIMARY_ACCESSTYPE: , QBLOCK_TYPE: SELECT
HINT_USED : , OPT_HINT_ID:
CORRELATION: , GROUP MEMB.: , COLLECTION : PMDEV,
VERSION : R710_LEVEL2
SORTN TABLE -UNIQUE: N, JOIN: N, ORDER BY: N, PARAL.GROUP: 0
SORTC TABLE -UNIQUE: N, JOIN: N, ORDER BY: Y, GROUP BY: N, PARAL.GROUP: 0
```

### Figure 64. Explain report – access path (for second statement step)

```
THE ACCESS PATH CHOSEN BY DB2 AT 11:41:47.5 ON 2008-09-05
```

---

*Figures 61, 62, 63, and 64 are part of a report's explanation section that detail host variables and statement details, including SQL queries and execution plans.*

---

84 Reporting User's Guide
Interpreting the Explain report

This section highlights some of the information to look for in an Explain report.

For a full discussion of the DB2 catalog information shown by the Explain report, see the DB2 Administration Guide and the DB2 SQL Reference.

The information reported varies depending on the type of report, the level of detail requested, and whether data is available to report. The example shown in the previous section shows an explain at package level. The package explained contains two SQL statements, the second statement has two statement steps (select and order).

Page header
The page header (Figure 43 on page 79) is printed at the top of each report page and shows general information about the report, including the type of report, subsystem identifier, DB2 version, and the objects reported.

Package version
The Package Version block of the report (Figure 44 on page 80) shows the package versions available and the versions and generations reported.

Statement details
The Statement Details block of the report (Figure 45 on page 80) shows the complete plan or package identification details, including:

- Location
- Name
- Statement number and text
- Other statement-related information
It also shows the status and isolation level, from SYSIBM.SYSPACKSTMT, and cost information. This information is repeated for each statement step.

**PLAN_TABLE information**
This shows the contents of the PLAN_TABLE (Figure 46 on page 80).

**Access path**
The Access Path block of the report (Figure 47 on page 81) shows, for each step in an SQL statement, the access path used by DB2 to satisfy the statement step. This shows:

**Use of index**
If an index is used, the type of index scan and details about matching columns are shown.

The appropriate use of indexes plays a key role in the efficiency of an application or query. For example, if a table contains 3000 rows of information in 100 data pages, the following query returns one row of data:

```
SELECT COL1 COL2 FROM T1 WHERE COL4=10 AND COL5=20
```

**No index**
If the table has no index associated, DB2 must scan each data page to satisfy the query. In this instance, DB2 must perform 100 Getpage operations for a single SELECT statement.

**Index on COL4**
In this instance, the number of Getpage operations depends on the number of rows matching the predicate and their distribution, as well as the number of index pages. For example, if rows matching the predicate are spread over six data pages and pointers are spread over two index pages, the number of Getpage operations for the SELECT is eight.

**Unique index on COL4 and COL5**
In this instance, DB2 only needs to read one page from the root index and one page from the leaf index to locate the data page. The number of Getpage operations for the SELECT is three.

**Index only scan**
When you create an index on COL1, COL2, COL4, and COL5, and alter the SELECT to predicate all four columns:

```
SELECT COL1 COL2 FROM T1 WHERE COL4=10 AND COL5=20
AND COL1 NOT NULL AND COL2 NOT NULL
```

In this instance, DB2 can satisfy the query from the index alone, requiring just one Getpage operation for the root index and one Getpage operation for the leaf index.

Properly organized indexes can also help reduce or eliminate sort operations.

You can define multiple indexes on a base table. However, weigh the performance gain with the additional workload required for the database manager to update the indexes when data changes in the table. Generally, multiple indexes are useful for tables that are often queried and seldom updated.
You can use the Accounting reports to derive the Getpage/SQL ratios for plans and packages. This is the total number of Getpage operations divided by the total number of SQL SELECT, INSERT, UPDATE, and FETCH statements. The value of the ratio depends on the type of application and the DB2 environment, but you might look for applications with ratios greater than five.

**Prefetch**

Prefetch improves the performance of DB2 by reducing the time spent waiting for sequential I/O. It can also substantially reduce the Getpage/SQL ratio.

For Sequential Prefetch, data must be ordered in the tables in the same way as it is accessed by the application. This allows DB2 to fetch the pages before they are accessed by the application. This means that the design and organization of tables and applications needs to be tailored to exploit this behavior. DB2 can also decide to use Sequential Prefetch if it sees that the data is sequenced. This is known as Dynamic Prefetch. For SQL queries DB2 can read up to 32 pages per Read I/O.

List Prefetch works very much like Sequential Prefetch, except that data pages do not need to be contiguous. List prefetch is always used for multiple index access and to obtain access to data from the inner table of a hybrid join.

**Direct row access**

Direct row access allows DB2 to obtain access to a row directly through the ROWID column. If an application selects a row from a table that contains a ROWID column, the row ID value implicitly contains the location of the row. If you use that row ID value in the search condition of subsequent SELECTs, DB2 might be able to navigate directly to the row.

To use direct row access, you first select the values of a row into host variables. The value that is selected from the ROWID column contains the location of that row. Later, when you perform queries which access that row, you include the row ID value in the search condition. If DB2 determines that it can use direct row access, DB2 uses the row ID value to navigate directly to the row.

Because direct row access provides highly efficient data access, consider modifying older applications to exploit this feature.

**Index data**

The Index Data block of the report (Figure 48 on page 81) shows data derived from the SYSIBM.SYSINDEXES table. This is only shown if an index is used in the access plan. You can use the Information in SYSINDEXES to compare the available indexes on a table to determine which one is the most efficient for a query.

**Key column data**

The Key Column Data block of the report (Figure 49 on page 81) shows index key information that is derived from the SYSIBM.SYSKEYS and SYSIBM.SYSCOLUMNS tables. This is only shown when an index is used in the access plan.

**Table**

The Table Data block of the report (Figure 50 on page 81) shows information that is derived from SYSIBM.SYSTABLES.
Table space
The Table Space Data block of the report (Figure 51 on page 81) shows information that is derived from SYSIBM.SYSTABLESPACE.

Host variables data
When host variables are used in a statement, OMEGAMON XE for DB2 PE can show details about each variable (Figure 52 on page 81). A host variable can be either a variable in a host language (such as a PL/I variable, C variable, Fortran variable, a COBOL data item, or Assembler language storage area) or a host language construct that was generated by an SQL precompiler from a variable declared using SQL extensions. A host variable can be an output value that is returned to the application by DB2 or an input to DB2.

Summary
The summary report (Figure 65 on page 85) shows the SQL statement number, type, and access path for each statement reported. When the summary is appended to a full report, the statement numbers are also cross-referenced to the report page where the full entry can be found.
Chapter 4. Other OMEGAMON XE for DB2 PE reports

Other reports are available to tune and maintain your DB2 system.

Other performance-related reports

The Locking Activity, I/O Activity, Utility Activity, and Record Trace reports can help you with specific known problems and for prototyping.

The following topics provide additional information:

- "Locking Activity reports"
- "I/O Activity reports" on page 94
- "Utility Activity reports" on page 97
- "Record Trace reports" on page 101

Locking Activity reports

The Locking report set provides various levels of detail about concurrency control within DB2 in the form of reports, traces, and a File data set.

It shows information about:

- DB2 transaction locks, which are locks on table spaces, tables, pages, or rows. DB2 transaction locks are used primarily to control access by SQL statements.
- DB2 drain locks and DB2 claims, which control access by DB2 utilities and commands.
- Global locks in a data sharing environment.
- DB2 lock avoidance techniques and related locking data, such as page latch waits.

Locking reports summarize all user activity related to lock suspensions and lockouts. The reported information is grouped and ordered by OMEGAMON XE for DB2 PE identifiers, which you can specify.

- A Suspension report is a summary of lock suspensions across the reporting interval for a unique combination of selected OMEGAMON XE for DB2 PE identifiers. The report also shows the number of occurrences and elapsed times, the causes, and the reasons for resuming processing.
- A Lockout report is a summary of users and resources involved in timeouts and deadlocks. For each event, the user holding the resource and the users waiting for the resource are identified, along with the number of occurrences and other statistics.

Locking traces provide information about lock events as they occur. Locking traces differ in the level of detail as follows:

- A Detail trace contains complete information about lock-related events in a DB2 system, or a group of DB2 systems in a data sharing environment. It shows all IMS/VS Internal Resource Lock Manager (IRLM) requests, IRLM suspensions, timeouts, deadlocks, claim and drain activities, lock-avoidance occurrences, and inter-DB2 requests in a data sharing environment.
- A Deadlock trace shows every occurrence of a deadlock.
- A Lockout trace shows each timeout and deadlock.
- A Suspension trace shows every lock suspension.
• A *Timeout trace* shows each timeout.

The following topics provide additional information:

• "When to use Locking reports"
• "Input for Locking reports"
• "Collecting Locking Activity data with a DB2 trace" on page 91
• "Creating an OMEGAMON XE for DB2 PE detail Locking trace" on page 91

**When to use Locking reports**

Use Locking reports when you need more information than can be supplied by the Accounting or Statistics reports and traces to resolve concurrency problems.

You can also use a Locking trace, for example, to monitor the effect of a new application introduced into a production system.

There is no correlation between the number of deadlock events shown in Locking reports and traces and the number of deadlocks shown in Accounting and Statistics reports. Accounting and Statistics reports count all deadlock occurrences regardless of how they resolve. Locking reports count only those deadlocks that were resolved by DB2. DB2 can resolve a deadlock either by making a process roll back, thereby releasing the locks it holds on resources, or by requesting a process to terminate.

**Input for Locking reports**

If you do not use the Collect Report Data facility supplied with the Online Monitor, you need to know the input IFCIDs for Locking reports.

The IFCIDs used as input for the Locking reports, traces, and File data sets are shown in the following table.

*Table 8. IFCIDs used for Locking reports*

<table>
<thead>
<tr>
<th>Locking report, trace, and File data set</th>
<th>IFCIDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock suspension report or trace</td>
<td>44, 45, 105, 107, 213, 214, 215, 216, 226, 227</td>
</tr>
<tr>
<td>Lockout report or trace</td>
<td>105, 107, 172, 196</td>
</tr>
<tr>
<td>Deadlock trace</td>
<td>105, 107, 172</td>
</tr>
<tr>
<td>Timeout trace</td>
<td>105, 107, 196</td>
</tr>
<tr>
<td>Lock detail trace</td>
<td>20, 21, 44, 45, 105, 107, 172, 196, 211, 212, 213, 214, 215, 216, 218, 223, 226, 227, 251, 257, 259, 337</td>
</tr>
<tr>
<td>File data set</td>
<td>21, 105, 107, 211, 212, 223</td>
</tr>
</tbody>
</table>

Deadlock traces and Timeout traces are generated from Statistics class 3. When Statistics class 3 is active, deadlock and Timeout traces can be generated without starting any Performance trace classes.

DB2 IFC covers all important aspects of DB2 locking activity provided that the appropriate trace classes are active. Except for IDENT, SYNC, or QUIT, each occurrence of an IRLM request is traced by an IFCID 21, 211, or 212 record.

To maintain consistency in a data sharing environment, DB2 IFC generates an IFCID 251, 257, or 259, which records, or notifies on, a physical lock (P-lock) on a page set, partition, or page.
Whenever a suspension occurs, DB2 IFC generates an IFCID 44, 213, 215, or 226 record along with a matching IFCID record that specifies the reason for resume, regardless of how the suspension was generated. The matching IFCID record is an IFCID 45 for IFCID 44, IFCID 214 for IFCID 213, IFCID 216 for IFCID 215, and IFCID 227 for IFCID 226. Also, the DB2 trace produces IFCID 211 records for every claim request and IFCID 212 records for every drain request.

An IFCID 172 record is written when lock suspensions end in deadlock or when deadlock situations occur. The deadlock record details all units of work involved in the deadlock, the resources for which they were contending, and the attributes of their requests.

IFCID 196 records lock suspensions that end in a timeout or when timeout situations occur. It details all units of work involved in the timeout, the resource for which they were contending, and the attributes of their requests.

IFCIDs 218 and 223 provide summary information and details of successful lock avoidance, which can improve application performance by reducing lock suspension times.

IFCID 20 records the locking summary for page or row locks, the highest table space lock state, and lock escalation information for table spaces. The record is written for the thread at each commit or at a rollback.

IFCID 105 and 107 provide database and object identifier translation information used in all Locking reports and traces and in Locking File.

The following example shows the DB2 trace commands that are required to collect data for a detail Locking report or trace:

```
-START TRACE(P) DEST(OPX) CLASS(6,17,7,30) IFCID(226,227,251,257,259) BUFSIZE(512)
-START TRACE(S) DEST(OPX) CLASS(3) BUFSIZE(512)
```

**Collecting Locking Activity data with a DB2 trace**

To create a Locking Activity report, you need to collect data by using a DB2 trace.

The easiest way to collect data for any OMEGAMON XE for DB2 PE report is to use the activate traces facility of the workstation monitor. You can use this facility to configure a DB2 trace and start and stop the trace from there. Using the activate traces facility also ensures that you include all data in the DB2 trace that is needed for a report. For more information, see *Monitoring Performance from Performance Expert Client*.

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE(P) DEST(OPX) CLASS(6,17,7,30) IFCID(226,227,251,257,259) BUFSIZE(512)
-START TRACE(S) DEST(OPX) CLASS(3) BUFSIZE(512)
```

**Creating an OMEGAMON XE for DB2 PE detail Locking trace**

Use your ISPF editor to create a JCL job that creates a detail Locking trace.

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.
To submit the job, type SUBMIT on the command line and press Enter.

The following figure shows an example of a detail Locking trace.
Figure 67. Example of a detail Locking trace
I/O Activity reports

DB2 system performance is heavily influenced by the volume and speed of input activities and output activities. OMEGAMON XE for DB2 PE provides several summary and detail I/O Activity reports that you can use to analyze I/O activities.

Generally, an I/O Activity report groups the I/O activities in report blocks as follows:

- The Active Log block shows the number of read and write requests to and from the active log data set and the associated wait times.
- The Archive Log block shows the number of read and write requests to and from the archive log and the associated wait times.
- The Bootstrap Data Set block shows the number of read and write requests to and from the bootstrap data set (if present). The bootstrap data set also controls the movement of data from the active log to the archive log data sets.
- The Buffer Pool block shows the number and types of read and write requests to and from the buffer pools and the associated wait times. It shows the volume of data pages that is moved between a hard disk drive and the main storage.
- The Cross-Invalidation Activity block shows the number of buffer refresh activities caused by cross-invalidation.

Cross-invalidation is necessary to maintain coherency of data within a DB2 data sharing group. It happens when a group member updates a data page and writes that page to the group buffer pool. All members that have this data page cached in their buffer pools are notified that the page was invalidated.
A high level of cross-invalidation affects performance because it effectively reduces the amount of buffer pool space available to the system. When an invalidated page is required, it must be refreshed in the buffer pool, either by retrieving the page from the group buffer pool or from a hard disk drive.

- The EDM Pool block shows the number of loads from a hard disk drive for cursor table (CT), package table (PT), and database directory (DBD) requests. For each event it shows the average size and load time.

The following topics provide additional information:

- "When to use I/O Activity reports"
- "Input for I/O Activity reports"
- "Collecting I/O Activity data with a DB2 trace"
- "Creating an OMEGAMON XE for DB2 PE I/O Activity trace" on page 96

**When to use I/O Activity reports**

Use I/O Activity reports if a Statistics report indicates a potential problem with logging, buffer pool or EDM pool activity, or high levels of cross-invalidation.

You can also run I/O Activity reports on a regular basis to gain an overview of the system and to help balance workloads between DB2 subsystems.

**Input for I/O Activity reports**

I/O Activity reports use IFCIDs from the DB2 Performance trace type, basically trace classes 4, 5, and 21.

The following table shows the IFCIDs used in I/O Activity reports.

<table>
<thead>
<tr>
<th>I/O Activity reports</th>
<th>IFCIDs used</th>
<th>DB2 trace type and trace class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Pool</td>
<td>6, 7, 8, 9, 10, 105, 107</td>
<td>Performance, class 4</td>
</tr>
<tr>
<td>EDM Pool</td>
<td>29, 30, 105, 107</td>
<td>Performance, class 4</td>
</tr>
<tr>
<td>Active Log</td>
<td>34, 35, 36, 37, 38, 39</td>
<td>Performance, class 5</td>
</tr>
<tr>
<td>Archive Log/BSDS</td>
<td>34, 35, 36, 37, 40, 41, 114, 115, 116, 119, 120</td>
<td>Performance, class 5</td>
</tr>
<tr>
<td>Cross-Invalidation</td>
<td>105, 107, 255</td>
<td>Performance, class 21</td>
</tr>
</tbody>
</table>

**Note:** If you want to include Sequential Prefetch Read I/O operations in the report, do not qualify the DB2 START TRACE command with a plan name or an authorization ID. Such a qualification for performance trace class 4 reduces the data DB2 puts in the user's task TCB and omits all asynchronous tasks including Sequential Prefetch.

**Collecting I/O Activity data with a DB2 trace**

To create an I/O Activity trace report, you need to collect data by using a DB2 trace.

The easiest way to collect data for any OMEGAMON XE for DB2 PE report is to use the activate traces facility of the workstation monitor. You can use this facility to configure a DB2 trace and start and stop the trace from there. Using the activate traces facility also ensures that you include all data in the DB2 trace that is needed for a report. For more information, see *Monitoring Performance from Performance Expert Client*.
Alternatively, you can start a DB2 trace as follows:

```
-START TRACE(P) DEST(OPX) CLASS(30)
IFCID(6,7,8,9,10,29,30,34,35,36,37,38,39,40,41,105,107,114,115,116,119,120,255)
BUFSIZE(512)
```

Creating an OMEGAMON XE for DB2 PE I/O Activity trace

Use your ISPF editor to create a JCL job that creates an I/O Activity trace.

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.

```
EDIT ---- ANDREW.OMPE.JOBS.IOSUMM--------------------- Columns 001 072
Command ===>

****** ************************************************ TOP OF DATA ******************************
0001 /* JCL to produce an I/O Activity Report */
0002 //ANOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 // STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.IOA,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.IOSUMM,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSGEN DD *
0010 IOACTIVITY
0011 REPORT
0012 EXEC
0013
****** ************************************************ BOTTOM OF DATA ******************************
```

Figure 69. I/O Activity job stream using ISPF/PDF editor

To submit the job, type SUBMIT on the command line and press Enter.

The following figure shows an example of a summary I/O Activity report.
Utility Activity reports

Many of the tasks of maintaining DB2 data, such as loading a table, copying a table space, or recovering a database to some previous point in time can be done using DB2 utilities. Utilities run as batch jobs under z/OS. The use of utilities, such as REORG INDEX, can significantly affect DB2 performance. You can use Utility Activity reports and traces to examine the efficiency of DB2 utilities.

Utility functions that are elements of separately orderable features of DB2 products must be licensed and installed in your environment to be properly reported. For information, see the *DB2 Utility Guide and Reference*.

Utility Activity reports and traces provide information about DB2 utility and bind activity during the processing of a DB2 application. Utility Activity reports show information as an aggregation of threads ordered by the combination of specified
OMEGAMON XE for DB2 PE identifiers. Utility Activity traces show information in chronological order. The information includes:

- Thread identification, thread start and stop time, and location
- Utility tools and bind statements executed, with the appropriate elapsed and TCB times
- LISTDEF information
- Utility data set information
- For trace only, optional workload detail, such as:
  - Bind activity
  - Exits
  - I/O activity
  - Lock suspensions
  - Page and row locking activity
  - Utility phases

OMEGAMON XE for DB2 PE can process data originating at different DB2 locations. In the JCL, you can logically concatenate multiple input data sets to a single data set that contains mixed records from multiple locations. If data from multiple locations is available, you can produce multi-site or single-site reports and traces.

- **Multi-site** reports and traces separate utility activity information according to the location where it occurs. Data is sequenced by location and includes activity initiated both locally and remotely.
- **Single-site** reports and traces show utility activity information from a single DB2 subsystem. They are produced from an input data set that contains data from a single site or, if the input data set contains data from multiple sites, by specifying a single location with the INCLUDE subcommand option or by suppressing locations with the EXCLUDE subcommand option.

Utility Activity reports and traces are logically grouped by bind activities and utility activities.

- Bind activity shows:
  - BIND
  - BIND PACKAGE
  - REBIND
  - REBIND PACKAGE
  - FREE PLAN
  - FREE PACKAGE
  - FREE REMOTE PACKAGE
  - BIND CONNECT
  - CONNECT RESET

- Utility activity

  Most utility events are comprised of detail events called phases. Each phase of the utility is reported. An event that does not have any detail events consists of one phase with the same name as the event. UTILINIT and UTILTERM phases are reported in a summary line. Each phase can have an item type.

The following topics provide additional information:

- "When to run Utility Activity reports" on page 99
- "Input for Utility Activity reports" on page 99
- "Collecting Utility Activity data with a DB2 trace" on page 100
- "Creating an OMEGAMON XE for DB2 PE Utility Activity trace" on page 100
**When to run Utility Activity reports**

Run Utility Activity reports as part of your regular monitoring policy or when you suspect problems, such as a high number of utility failures.

**Input for Utility Activity reports**

Utility Activity reports use IFCIDs from the DB2 Accounting and Performance trace types.

The following table shows the utility events that can be collected, together with the trace classes and IFCIDs used.

*Table 10. IFCIDs used for Utility Activity reports*

<table>
<thead>
<tr>
<th>DB2 trace type</th>
<th>DB2 trace class</th>
<th>DB2 IFCID</th>
<th>Record type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting</td>
<td>1</td>
<td>3</td>
<td>Accounting data</td>
</tr>
<tr>
<td>Bind Events</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>3</td>
<td>22</td>
<td>Minibinds generated by DB2 at bind prepare time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63</td>
<td>SQL statement to be parsed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>177</td>
<td>Package allocation</td>
</tr>
<tr>
<td>Performance</td>
<td>10</td>
<td>108</td>
<td>Begin bind or rebinding plan/package</td>
</tr>
<tr>
<td></td>
<td></td>
<td>109</td>
<td>End bind or rebinding plan/package</td>
</tr>
<tr>
<td></td>
<td></td>
<td>110</td>
<td>Begin free plan/package</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111</td>
<td>End free plan/package</td>
</tr>
<tr>
<td>Performance</td>
<td>16</td>
<td>183</td>
<td>Requesting agent data</td>
</tr>
<tr>
<td>Utilities</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>10</td>
<td>23</td>
<td>Start utility run, start of subtask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24</td>
<td>Utility change; phase info, subtask info</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>Utility end</td>
</tr>
<tr>
<td></td>
<td></td>
<td>219</td>
<td>LISTDEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>220</td>
<td>Data set info</td>
</tr>
<tr>
<td>I/O Events</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>4</td>
<td>6</td>
<td>Begin Read I/O, data set on DASD to buffer pool</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>End Read I/O, data set on DASD to buffer pool</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>Begin synchronous Write I/O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>Write end</td>
</tr>
<tr>
<td></td>
<td></td>
<td>226</td>
<td>Page latch wait begin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>227</td>
<td>Page latch wait end</td>
</tr>
<tr>
<td>Lock Suspension Events</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>6</td>
<td>44</td>
<td>Lock suspension or an identity call to the IRLM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>Lock resumption</td>
</tr>
<tr>
<td></td>
<td></td>
<td>213</td>
<td>Beginning of a wait for a drain lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>214</td>
<td>End of a wait for a drain lock</td>
</tr>
</tbody>
</table>
Table 10. IFCIDs used for Utility Activity reports (continued)

<table>
<thead>
<tr>
<th>DB2 trace type</th>
<th>DB2 trace class</th>
<th>DB2 IFCID</th>
<th>Record type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>17</td>
<td>213</td>
<td>Beginning of a wait for a drain lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>214</td>
<td>End of a wait for a drain lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>215</td>
<td>Begin of a wait for the claim count to go to zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>216</td>
<td>End of a wait for the claim count to go to zero</td>
</tr>
<tr>
<td>Page and Row Locking Events</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>6</td>
<td>20</td>
<td>Page and row locking summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>218</td>
<td>Lock avoidance summary</td>
</tr>
<tr>
<td>Exit Events</td>
<td></td>
<td>11</td>
<td>Validation exit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>Encode edit exit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19</td>
<td>Decode edit exit</td>
</tr>
</tbody>
</table>

**Collecting Utility Activity data with a DB2 trace**

To create an Utility Activity report, you need to collect data by using a DB2 trace.

The easiest way to collect data for any OMEGAMON XE for DB2 PE report is to use the activate traces facility of the workstation monitor. You can use this facility to configure a DB2 trace and start and stop the trace from there. Using the activate traces facility also ensures that you include all data in the DB2 trace that is needed for a report. For more information, see *Monitoring Performance from Performance Expert Client*.

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE(A) DEST(OPX) CLASS(1, 2, 3, 5, 7, 8) BUFSIZE(512)
-START TRACE(P) DEST(OPX) CLASS(13,30)
  IFCID(6,7,8,9,10,20,22,23,25,44,45,63,108,109,110,111,177,183,213,214,215,216,
  218,226,227) BUFSIZE(512)
-START TRACE(AU) DEST(OPX) CLASS(8) BUFSIZE(512)
```

**Creating an OMEGAMON XE for DB2 PE Utility Activity trace**

Use your ISPF editor to create a JCL job that creates a Utility Activity trace.

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.
The following figure shows an example trace produced from this job.

```
EDIT ---- ANDREW.OMPE.JOBS.UTTRA--------------------- Columns 001 072
Command ===>
****** *********************** TOP OF DATA ***********************
0001 /* JCL to produce an Utility Activity Report */
0002 //AWOMPEA JOB (TTS1,YUS7), 'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.UTI,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.UTTRAC,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 UTILITY
0011 TRACE
0012 ORDER (PRIMAUTH-PLANNAME-INSTANCE)
0013 EXEC
0014 ****** *********************** BOTTOM OF DATA ***********************
```

Figure 71. Utility Activity job stream using ISPF/PDF editor

The following figure shows an example trace produced from this job.

```
UTILITY TRACE COMPLETE
```

Figure 72. Example of a short Utility Activity trace

**Record Trace reports**

Record Trace reports show IFCID information as presented by DB2.

This IFCID information is used to produce other OMEGAMON XE for DB2 PE reports. The difference is that for other reports, this information can be interpreted, manipulated, or not included.

- A *summary* Record Trace report lists all selected records, together with a description, but without any record data. You can use this listing to determine what events occurred during the DB2 trace.
A short Record Trace report presents nonserviceability data from selected records that appear on other OMEGAMON XE for DB2 PE reports. Some large IFCIDs (for example, system statistics) are presented similar to summary Record traces, without record data.

A long Record Trace report presents serviceability and nonserviceability data from selected records.

A dump Record Trace report presents selected records in the standard hexadecimal dump format. You can use the dump Record Trace report to view data that might contain unprintable characters.

The following topics provide additional information:
- "When to use Record Trace reports"
- "Input for Record Trace reports"
- "Collecting Record Trace data with a DB2 trace" on page 104
- "Creating an OMEGAMON XE for DB2 PE Record Trace report" on page 105

**When to use Record Trace reports**
Use Record Trace reports if the information in other reports do not provide the required level of detail.

**Input for Record Trace reports**
Record Trace reports use IFCIDs from several DB2 trace types and trace classes.

Input to the Record Trace report set consists of all types of DB2 instrumentation data. The DB2 trace types and classes used by Record Trace reports are listed in Table 11.

<table>
<thead>
<tr>
<th>DB2 trace type</th>
<th>DB2 trace class</th>
<th>DB2 IFCIDs used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics</td>
<td>1</td>
<td>1, 2, 105, 106, 202</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>172, 196, 250, 261, 262, 313</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>191, 192, 193, 194, 195, 203, 204, 205, 206, 207, 208, 209, 210, 235, 236, 238, 267, 268</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>230, 254</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>199</td>
</tr>
<tr>
<td>Accounting</td>
<td>1</td>
<td>3, 106, 239</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>232</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6, 7, 8, 9, 32, 33, 44, 45, 117, 118, 127, 128, 170, 171, 174, 175, 213, 214, 215, 216, 226, 227, 242, 243</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>151</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>187</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>232, 240</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>6, 7, 8, 9, 32, 33, 44, 45, 117, 118, 127, 128, 170, 171, 174, 175, 213, 214, 215, 216, 226, 227, 241, 242, 243</td>
</tr>
</tbody>
</table>
Table 11. IFCIDs used for Record Trace reports (continued)

<table>
<thead>
<tr>
<th>DB2 trace type</th>
<th>DB2 trace class</th>
<th>DB2 IFCIDs used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit</td>
<td>1</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>55, 83, 87, 169, 312</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>23, 24, 25</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>146</td>
</tr>
<tr>
<td>Monitor</td>
<td>1</td>
<td>1, 2, 106, 124, 129, 147, 148, 149, 150, 202, 230, 254, 306</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>232</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6, 7, 8, 9, 32, 33, 44, 45, 51, 52, 56, 57, 117, 118, 127, 128, 170, 171, 174, 175, 213, 214, 215, 216, 226, 227, 242, 243</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>187</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>185</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>232, 240</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>6, 7, 8, 9, 32, 33, 44, 45, 51, 52, 56, 57, 117, 118, 127, 128, 170, 171, 174, 175, 213, 214, 215, 216, 226, 227, 241, 242, 243</td>
</tr>
</tbody>
</table>
Table 11. IFCIDs used for Record Trace reports (continued)

<table>
<thead>
<tr>
<th>DB2 trace type</th>
<th>DB2 trace class</th>
<th>DB2 IFCIDs used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>1</td>
<td>1, 2, 31, 42, 43, 76, 77, 78, 79, 102, 103, 105, 106, 107, 153</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3, 6, 8, 9, 10, 29, 30, 105, 106, 107, 108, 109, 110, 111, 201, 256</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 104, 105, 106, 114, 115, 116, 117, 118, 119, 120, (228, 229)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6, 7, 8, 9, 10, 105, 106, 107, 127, 128, 226, 227</td>
</tr>
<tr>
<td>Global</td>
<td>1</td>
<td>106, (132, 134, 138)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>106, (131, 133, 139)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0, 38, 46, 47, 48, 49, 50, 51, 52, 56, 57, 58, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 93, 94, 106, 114, 115, 116, 117, 118, 119, 120, (228, 229)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>106, (130)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>190, 249 (135, 136, 137,) (247, 248)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>156</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>164, 165, 166</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>180, 181, 182</td>
</tr>
</tbody>
</table>

**Collecting Record Trace data with a DB2 trace**

To create a Record Trace report, you need to collect data by using a DB2 trace.
A Record trace can be very long. Generally, run a Record trace on one component at a time to focus on a problem. The following example collects records associated with Accounting.

The easiest way to collect data for any OMEGAMON XE for DB2 PE report is to use the activate traces facility of the workstation monitor. You can use this facility to configure a DB2 trace and start and stop the trace from there. Using the activate traces facility also ensures that you include all data in the DB2 trace that is needed for a report. For more information, see Monitoring Performance from Performance Expert Client.

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE(A) DEST(OPX) CLASS(1, 2, 3, 5, 7, 8) BUFSIZE(512)
-START TRACE(P) DEST(OPX) CLASS(17, 16, 14, 13, 12, 11, 10, 9, 8, 7, 4, 22, 21, 20, 30)
IFCID(20, 22, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 44, 45, 53, 55, 58, 59, 60, 61,
62, 63, 64, 65, 66, 76, 77, 78, 79, 80, 81, 92, 97, 103, 104, 112, 114, 115, 116, 117, 118,
119, 120, 124, 129, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154,
155, 169, 170, 171, 177, 178, 179, 186, 188, 191, 192, 193, 194, 195, 198, 203, 204, 205, 206,
207, 208, 209, 210, 218, 219, 220, 224, 228, 229, 233, 235, 236, 237, 238, 239, 243, 247, 254,
258, 272, 273, 312, 313, 314, 324, 325, 329, 330, 331, 332, 333, 334) BUFSIZE(512)
-START TRACE(S) DEST(OPX) CLASS(6, 5, 3, 1) BUFSIZE(512)
```

Creating an OMEGAMON XE for DB2 PE Record Trace report

Use your ISPF editor to create a JCL job that creates a Record Trace report.

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.

```
EDIT ---- ANDREW.OMPE.JOBS.RECTRCE--------------------- Columns 001 072
Command ===> ________________________________________________ Scroll ===> CSR_
****** ***************************** TOP OF DATA *****************************
0001 /* JCL to produce an SQL activity trace */
0002 //AWOMPEA JOB (TTS1,YUS7), 'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //SYSIN DD *
0007 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE2,DISP=SHR
0008 //OUTPUTDD DD DSN=ANDREW.OMPE.OUT.REC01,DISP=OVR
0009 //JOBSUMDD DD SYSOUT=A
0010 RECTRACE
0011 LEVEL (LONG)
0012 SUMMARIZEBY (TIMESTAMP)
0013 EXEC
****** ***************************** BOTTOM OF DATA *****************************
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel
Figure 73. Record trace job stream using ISPF/PDF editor
```

To submit the job, type SUBMIT on the command line and press Enter.

Housekeeping reports

These reports are not strictly performance reports. They present information about the DB2 system and user activity.

They provide valuable information, when used together with Accounting and Statistics reports, and help to gain an overview of the DB2 system.

The following topics provide additional information:
The Audit report set

With DB2 Audit data you can track DB2 resource access. Audit reports and traces show information about the user of an auditable object and the time and type of action performed on the object.

Audit reports group events in the order of event type, ORDER identifier, and timestamp.

Audit traces show events listed chronologically.

You can use Audit reports and traces to review Audit data by OMEGamon XE for DB2 PE identifiers and choose between a wide range of options for presenting DB2 Audit data.

You can order Audit reports and traces by the object of the Audit event (for example, authorization failures per table).

You can produce member-scope and group-scope Audit reports and traces.

- Member-scope reporting presents data member by member.
- Group-scope reporting merges the data that is produced by each group member to give a full picture of the use of a shared resource.

You can produce Audit reports and traces of individual Audit events in the order in which they occurred, or reports of aggregated Audit data.

You can specify the level of detail in Audit reports and traces by specifying any combination of the following event types:

- Authorization failures
- Changes to authorization identifiers
- DDL operations against auditable tables
- DML statements at bind of auditable tables
- Grants or revokes of privileges
- Read/write access to auditable tables
- Utility access to auditable tables

The following topics provide additional information:

- "When to use Audit reports"
- "Collecting Audit data with a DB2 trace" on page 107
- "Creating an OMEGamon XE for DB2 PE Audit trace" on page 107

When to use Audit reports

Use Audit reports or traces regularly to assist in your database administration and as part of your regular monitoring policy. More specifically, use the Audit reports if Accounting reports show unexpected numbers of authorization failures.

You can use Audit reports to monitor:

Usage of sensitive data

Tables that contain sensitive data, such as employee salary records, should probably be defined with AUDIT ALL. You can report usage by table and by
authorization ID to look for access by unusual IDs, at unusual times, or of unexpected types. You also want to record any ALTER or DROP operations that affect the data.

**Grants of critical privileges**
Authorities such as SYSADM and DBADM and explicit privileges over sensitive data, such as an Update privilege on records of accounts payable, must be monitored carefully. A query of the DB2 catalog can show who holds such a privilege at a particular time. The Audit records can reveal whether the privilege was granted and then revoked in a period of time.

**Unsuccessful access attempts**
Some unsuccessful access attempts are only user errors, but others can be attempts to violate security. All must be investigated. If you have sensitive data, always use Audit class 1 trace data.

**Collecting Audit data with a DB2 trace**
To create an Audit report, you need to collect data by using a DB2 trace.

The easiest way to collect data for any OMEGAMON XE for DB2 PE report is to use the activate traces facility of the workstation monitor. You can use this facility to configure a DB2 trace and start and stop the trace from there. Using the activate traces facility also ensures that you include all data in the DB2 trace that is needed for a report. For more information, see *Monitoring Performance from Performance Expert Client*.

Alternatively, you can start a DB2 trace as follows:

```-START TRACE(AU) DEST(OPX) CLASS(1,2,3,4,5,6) BUFSIZE(512)
-START TRACE(P) DEST(OPX) CLASS(30) IFCID(24,55,83,87,105,107,169) BUFSIZE(512)```

**Creating an OMEGAMON XE for DB2 PE Audit trace**
Use your ISPF editor to create a JCL job that creates an Audit trace.

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.

```
EDIT ---- ANDREW.OMPE.JOBS.AUDTRC--------------------- Columns 001 072
Command ==> ___________________________________________ Scroll ==> CSR_
****** ******************************************* TOP OF DATA *******************************************
0001 /* JCL to produce an Audit trace */
0002 //AWOMPEA JOB (TTS1,YUS7), 'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.AUDIT,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.AUDT1,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 AUDIT
0011 TRACE
0012 SCOPE (MEMBER)
0013 EXEC
****** ******************************************* BOTTOM OF DATA *******************************************
F1=Help  F2=Split  F3=Exit  F5=Rfind  F6=Rchange  F7=Up
F8=Down  F9=Swap  F10=Left  F11=Right  F12=Cancel
```

*Figure 74. Audit job stream using ISPF/PDF editor*

To submit the job, type SUBMIT on the command line and press Enter.
The following figure shows an example of the Audit trace produced.

Figure 75. Example of a member-scope Audit trace

The System Parameters report

This report provides information about the configuration of the DB2 system being monitored.

A report entry is produced for each location present in the input data. An entry is also produced if DB2 was restarted with changed system parameters or a change to the system parameters was detected when the Statistics interval was reached.

Some parameters, such as buffer pool and group buffer pool attributes can be changed while a system is active. If the appropriate DB2 trace class is active, the changes are recorded in the System Parameters report in the order of occurrence.

The following topics provide additional information:

- "When to use System Parameters reports" on page 109
- "Collecting System Parameters data with a DB2 trace" on page 109
- "Creating a System Parameters report" on page 109
When to use System Parameters reports
Use the System Parameters report regularly as part of your system monitoring policy.

You can also use this report together with a Statistics report to determine whether you can improve a system's performance by altering its system settings.

Collecting System Parameters data with a DB2 trace
To create a System Parameters report, you need to collect data by using a DB2 trace.

The easiest way to collect data for any OMEGAMON XE for DB2 PE report is to use the activate traces facility of the workstation monitor. You can use this facility to configure a DB2 trace and start and stop the trace from there. Using the activate traces facility also ensures that you include all data in the DB2 trace that is needed for a report. For more information, see *Monitoring Performance from Performance Expert Client*.

Alternatively, you can start a DB2 trace as follows:
-START TRACE(S) DEST(OPX) CLASS(5) BUFSIZE(512)
-START TRACE(P) DEST(OPX) CLASS(30) IFCID(106,201,202,256) BUFSIZE(512)

Creating a System Parameters report
Use your ISPF editor to create a JCL job that creates a System Parameters report.

The following JCL is an example. Remember to change the setup information (starting with // in your JCL) as appropriate for your installation.

```r
EDIT ---- ANDREW.OMPE.JOBS.SYPTRC--------------------- Columns 001 072
Command ===> ________________________________________________ Scroll ===> CSR_
****** *********************** TOP OF DATA ******************************
0001 /* JCL to produce a System Parameters report */
0002 //AWOMPEA JOB (TTS1,YUS7),ANDREW,'
0003 // MSGCLASS=V,CLASS=O,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.SYSP,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.SYSPT1,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 SYSPARMS
0011 EXEC
0012 **** *********************** BOTTOM OF DATA ************************
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel
```

Figure 76. System parameters job stream using ISPF/PDF editor

To submit the job, type `SUBMIT` on the command line and press Enter.

The following figure shows an example of the System Parameters report produced.
Figure 77. Example of the System Parameters report
### BUFFER POOL PARAMETERS

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>BUFFER POOL ID</th>
<th>VIRTUAL POOL TYPE</th>
<th>VIRTUAL POOL SIZE (PAGES)</th>
<th>HIPERPOOL SIZE (PAGES)</th>
<th>VERTICAL POOL SEQUENTIAL THRESHOLD</th>
<th>HORIZONTAL DEFERRED WRITE THRESHOLD</th>
<th>VERTICAL DEFERRED WRITE THRESHOLD (PERCENTAGE)</th>
<th>VERTICAL DEFERRED WRITE THRESHOLD (BUFFERS)</th>
<th>ASSISTING PARALLEL SEQUENTIAL THRESHOLD</th>
<th>PFFFFFFX ATTRIBUTE</th>
<th>PAGE STEAL METHOD</th>
<th>AUTOSIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/20/08 08:44:53.00</td>
<td>BP0</td>
<td>N/A</td>
<td>20000</td>
<td>N/A</td>
<td>80</td>
<td>30</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>NO</td>
<td>LRU</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/20/08 08:44:53.00</td>
<td>BP1</td>
<td>N/A</td>
<td>20000</td>
<td>N/A</td>
<td>80</td>
<td>30</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>NO</td>
<td>LRU</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/20/08 08:44:53.00</td>
<td>BP2</td>
<td>N/A</td>
<td>20000</td>
<td>N/A</td>
<td>80</td>
<td>30</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>NO</td>
<td>LRU</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 78. Example of the System Parameters report (continued)
### Buffer Pool Parameters

<table>
<thead>
<tr>
<th>Buffer Pool ID</th>
<th>Virtual Pool Size (Pages)</th>
<th>Virtual Pool Sequential Threshold</th>
<th>Hiperpool Sequential Threshold</th>
<th>Hiperpool Size (Pages)</th>
<th>Virtual Pool Parallel Sequential Threshold</th>
<th>Assisting Parallel Sequential Threshold</th>
<th>Pgfix Attribute</th>
<th>Page Steal Method</th>
<th>Autosize</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP7</td>
<td>1000</td>
<td>80</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>50</td>
<td>No</td>
<td>LRU</td>
<td>No</td>
</tr>
<tr>
<td>BP32K</td>
<td>250</td>
<td>80</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>50</td>
<td>No</td>
<td>LRU</td>
<td>No</td>
</tr>
<tr>
<td>BP32K7</td>
<td>250</td>
<td>80</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>50</td>
<td>No</td>
<td>LRU</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 79. Example of the System Parameters report (continued)
BUFFER POOL PARAMETERS

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>10/20/08 08:44:53.00</th>
<th>VIRTUAL POOL ID</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFER POOL ID</td>
<td>BP8K0</td>
<td>VIRTUAL POOL TYPE</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VIRTUAL POOL SIZE (PAGES)</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hiperpool size (PAGES)</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Virtual pool sequential threshold</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hiperpool sequential threshold</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Horizontal deferred write threshold</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vertical deferred write threshold (percentage)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vertical deferred write threshold (buffers)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Virtual pool parallel sequential threshold</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assisting parallel sequential threshold</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pgfxt attribute</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page steal method</td>
<td>LRU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Autosize</td>
<td>NO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>10/20/08 08:44:53.00</th>
<th>VIRTUAL POOL ID</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFER POOL ID</td>
<td>BP16K0</td>
<td>VIRTUAL POOL TYPE</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VIRTUAL POOL SIZE (PAGES)</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hiperpool size (PAGES)</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Virtual pool sequential threshold</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hiperpool sequential threshold</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Horizontal deferred write threshold</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vertical deferred write threshold (percentage)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vertical deferred write threshold (buffers)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Virtual pool parallel sequential threshold</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assisting parallel sequential threshold</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pgfxt attribute</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page steal method</td>
<td>LRU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Autosize</td>
<td>NO</td>
</tr>
</tbody>
</table>

Figure 80. Example of the System Parameters report (continued)
Figure 81. Example of the System Parameters report (continued)
Figure 82. Example of the System Parameters report (continued)
Figure 83. Example of the System Parameters report (continued)
### Other System Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUAL BSDS MODE (TWOBSDS)</td>
<td>YES</td>
</tr>
<tr>
<td>ROLL UP PARALLEL TASK ACCOUNTING (PTASKROL)</td>
<td>YES</td>
</tr>
<tr>
<td>NO. PAGES SMALL TABLE THRESHOLD (NPSTHRESH)</td>
<td>0</td>
</tr>
<tr>
<td>SMS DATACLASS NAME FOR TS (SMSDCFL)</td>
<td>N/P</td>
</tr>
<tr>
<td>SMS DATACLASS NAME FOR IS (SMSDCIX)</td>
<td>N/P</td>
</tr>
<tr>
<td>COMPATIBILITY OPTION (COMPAT)</td>
<td>N/A</td>
</tr>
<tr>
<td>OFFLOAD OPTION (OFFLOAD)</td>
<td>YES</td>
</tr>
<tr>
<td>SU CONVERSION FACTOR</td>
<td>.818</td>
</tr>
<tr>
<td>OUTER JOIN PERFORMANCE ENHANCEMENTS (OJPERFEH)</td>
<td>YES</td>
</tr>
<tr>
<td>MINIMUM DIVIDE SCALE (MINDIVSCL)</td>
<td>NONE</td>
</tr>
<tr>
<td>STAR JOIN THRESHOLD (SJTABLES)</td>
<td>10</td>
</tr>
<tr>
<td>MVS ENVIRONMENT</td>
<td>XA</td>
</tr>
<tr>
<td>ONLINE SYSTEM PARM USER ID MONITOR</td>
<td>N/P</td>
</tr>
<tr>
<td>ONLINE SYSTEM PARM CORREL ID MONITOR</td>
<td>N/P</td>
</tr>
<tr>
<td>ONLINE SYSTEM PARM TIME CHANGED</td>
<td>N/P</td>
</tr>
<tr>
<td>ONLINE SYSTEM PARM TYPE</td>
<td>X'100'</td>
</tr>
<tr>
<td>DB2-SUPPLIED DECP INDICATOR</td>
<td>X'D5'</td>
</tr>
<tr>
<td>MAX CONCURRENT PKG OPS (MAX_CONCURRENT_PKG_OPS)</td>
<td>10</td>
</tr>
<tr>
<td>MAX TEMP STORAGE PER AGENT IN MB (MAXTEMPS)</td>
<td>0</td>
</tr>
<tr>
<td>ADMIN SCHEDULER JCL PROC NAME (ADMTPROC)</td>
<td></td>
</tr>
<tr>
<td>FREE LOCAL CACHED STATEMENTS (CACHEDYN_FREELocal)</td>
<td>0</td>
</tr>
<tr>
<td>DB2 UTILITIES PARAMETERS (DSNTIP6)</td>
<td></td>
</tr>
<tr>
<td>SYSTEM-LEVEL BACKUPS (SYSTEM_LEVEL_BACKUPS)</td>
<td>NO</td>
</tr>
<tr>
<td>RESTORE/RECOVER (RESTORE_RECOVER_FROMDUMP)</td>
<td>NO</td>
</tr>
<tr>
<td>DUMP CLASS NAME (UTILS_DUMP_CLASS_NAME)</td>
<td>'BLANK'</td>
</tr>
<tr>
<td>MAXIMUM TAPE UNITS (RESTORE_TAPEUNITS)</td>
<td>0</td>
</tr>
<tr>
<td>TEMPORARY UNIT NAME (VOLTDEVT)</td>
<td>SYSDA</td>
</tr>
<tr>
<td>UTILITY CACHE OPTION (SEQPRES)</td>
<td>NO</td>
</tr>
<tr>
<td>STATISTICS HISTORY (STATHIST)</td>
<td>NO</td>
</tr>
<tr>
<td>STATISTICS ROLLUP (STATROLL)</td>
<td>NO</td>
</tr>
<tr>
<td>STATISTICS CLUSTERING (STATCLUS)</td>
<td>ENHANCED</td>
</tr>
<tr>
<td>UTILITY TIMEOUT FACTOR (UTIMOUT)</td>
<td>6</td>
</tr>
</tbody>
</table>

### Databases and Spaces Started Automatically (DSNTIPS)

ALL

ICF Catalog Qualifiers

-------------

DSN911

Sizes Panel 1 (DSNTIPS)

-------------

USER LOB VALUE STORAGE IN KB (LOBVALA).................10,240
SYSTEM LOB VALUE STORAGE IN MB (LOBVALS)..............2,048
MAXIMUM NUMBER OF LE TOKENS (LEMAX)...................20
USER XML VALUE STG IN KB (XMLVALA)....................1,024,768
SYSTEM XML STG IN MB (XMLVALS).........................2,048

Figure 84. Example of the System Parameters report (continued)
Chapter 5. Advanced reporting concepts

The following information describes advanced and more detailed reporting concepts.

Accounting report set

This topic describes the Accounting report set and concepts in detail.

From the data shown in Accounting reports you can assign DB2 resource costs to individual authorization IDs and tune programs. OMEGAMON XE for DB2 PE does not, however, provide for tasks such as charge-back or billing.

Typical uses of Accounting reports are to:

- Summarize the local and distributed DB2 activity associated with the execution of specified DB2 plans.
- Summarize the composite DB2 activity associated with the execution of threads that exploit query CP parallelism or Sysplex query parallelism.
- Summarize the DB2 activity associated with the execution of the specified DB2 packages and DBRMs.
- Detect potential problem areas within DB2 applications.
- Track DB2 resource usage on a regular basis by OMEGAMON XE for DB2 PE identifiers such as location, authorization ID, or plan name. You can use this information to study trends or deviations from trends.
- Identify DB2 threads that fail to meet user-specified criteria.

In data sharing groups, you can produce Accounting reports at member level or at group level.

Accounting reports provide summary data about DB2 resource usage for:

- A given thread
- In thread reuse, the interval of time between two signons
- All tasks within a thread that exploit query CP or Sysplex query parallelism

Exception processing is supported in the Accounting report set. You define exception thresholds for certain Accounting fields. When you request exception processing, the values in these fields are checked against the thresholds. Only those records that contain field values outside the defined thresholds are reported.

TOP processing identifies the main consumers of DB2 resources. If the TOP subcommand option is used, an index-like report is generated that shows the main consumers at the end of the report or trace. If you only want to see the main resource consumers, specify the TOP subcommand option with the ONLY keyword.

The following topics provide additional information:

- General Accounting information
- "Creating effective Accounting reports" on page 129

General Accounting information

This topic provides detailed descriptions of functions and parameters common to the Accounting report set.
Functions of the Accounting report set

The Accounting report set consists of the REPORT, TRACE, FILE, REDUCE, SAVE, and RESTORE functions.

REPORT

Shows Accounting data summarized by OMEGAMON XE for DB2 PE identifiers. For example, you can produce a report that shows the Accounting information for threads, plans, or users. The report shows totals and subtotals of fields.

For a DB2 system in a data sharing group, reports can be at member or group level.

TRACE

Shows Accounting data for a particular thread. Unlike reports, traces show the Accounting data without aggregation. This means that the records are listed individually, in the order of occurrence. There is an exception if a thread exploits parallelism. In this instance, all parallel activity is aggregated and presented as a single trace entry.

Traces can be very long. Use traces only to resolve a specific problem.

FILE

Formats DB2 Accounting records and stores the records in sequential data sets that can be loaded into DB2 tables. You can analyze the data in DB2 tables and produce tailored reports by using a reporting facility such as Query Management Facility (QMF). You can also use FILE to produce data sets that contain only exception records.

REDUCE

Reduces the volume of data that is input to the REPORT and SAVE functions. REDUCE is invoked automatically if you use REPORT or SAVE. Specify REDUCE explicitly if you want to:

- Specify an interval to order data on Accounting reports
- Produce several reports to cover different time periods

SAVE

Produces VSAM data sets that contain reduced Accounting records. When the data is saved, you can:

- Combine it with new data to produce long-term reports
- Use the Save-File utility to create sequential data sets that are suitable for use by the DB2 LOAD utility

You can use REDUCE and SAVE to keep historical Accounting data about DB2 performance. You can define the interval and the input filters for the data that is saved into the Save data set. After reducing data, the resulting data set is much smaller than the original input data set. However, reducing data uses a considerable amount of system resources.

RESTORE

Reloads a previously saved data set for additional use. Saved data can be restored and resaved as often as required.

Thread types

The DB2 thread or, for a thread in reuse, the part of it that is between two consecutive signons or resignon, is the basic unit of reporting for the Accounting report set.

OMEGAMON XE for DB2 PE uses the following categorization of DB2 threads:
**Allied thread**
An allied thread does not involve Distributed Data Facility (DDF) activity. It is initiated by local location and does not request data from another location. The Accounting record that represents an allied thread consists of the following set of data. This set of data is called *non-DDF data*.
- Identification of the thread
- General timing
- SQL and RID list usage
- Query parallelism
- Buffer pool activity
- Group buffer pool activity
- Data sharing locking
- Stored procedures
- Data capture
- Locking activity
- Packages and DBRMs executed
- Resource limit facility data

**Allied-distributed thread**
An allied-distributed thread is not initiated by a remote location, but requests data from one or more server locations. The Accounting record that represents an allied-distributed thread consists of:
- Non-DDF data
- One block of *DDF data* for each participating server location
DDF data includes such information as the number of messages, statements, rows, and bytes that have been sent and received.

**Database access thread (DBAT)**
A DBAT is initiated, created, and performed by a thread on behalf of a remote (requester) location. The Accounting record that represents a DBAT consists of:
- Non-DDF data
- DDF data for the requester location

**DBAT-distributed thread**
A DBAT-distributed thread is initiated by a requester location that in turn requests data from another server location.

For example, when location A uses DRDA to request data at location B and, in the same unit of work, accesses data at location C (using DB2 private protocol), the thread created at location A is an allied-distributed thread, the thread created at location B is a DBAT-distributed thread. The thread created at location C is a DBAT.

The Accounting record that represents a DBAT-distributed thread consists of:
- Non-DDF data
- DDF data for the requester location
- One block of DDF data for each participating server location

The following terms can help you to understand the concepts of the different thread types and merged processing:

**Nondistributed transaction**
A transaction that is initiated by DB2 and performed at one location without interaction with other locations.
For example, if an allied thread is not reused, it represents a nondistributed transaction. If it is reused, a nondistributed transaction is DB2 activity between two signons.

**Distributed transaction**
A transaction that is initiated by DB2 at one (requester) location and performed at one or more remote (server) locations.

Distributed transactions consist of local activity that is represented by an allied-distributed thread, and in case of a loopback from a DBAT, remote activity that is represented by one or more DBATs. Therefore, distributed transaction requires Accounting records for the allied-distributed thread and all corresponding DBATs.

Reports and traces are location-oriented. They show activity that is performed at one or more locations. For a given location, the following information is shown:

- The nondistributed transactions, in other words, the allied threads at that location
- The local activity of distributed transactions that originate at that location, in other words, the allied-distributed threads from that location without the corresponding DBATs at other locations
- The remote activity at that location as part of distributed transactions requested from other locations, in other words, the DBATs at that location

Reports and traces can be single-site or multi-site:

- Single-site reports and traces present Accounting data for one location. You can obtain a single-site report or trace by processing input data that only contains records from a single location or by specifying a single location with the INCLUDE or EXCLUDE subcommand options.
- Multi-site reports and traces present Accounting data for more than one location. The data is arranged in alphabetical order by location name.

**Distributed activity**
The Accounting trace provides insights to activities among distributed DB2 subsystems as well as non-DB2 requestors.

OMEGAMON XE for DB2 PE supports communication between:

- DB2 for z/OS subsystems
- DB2 for z/OS subsystem and non-DB2 requesters, such as SQL/DS™, or ORACLE
- DB2 for z/OS subsystem and DB2 for multiplatforms

For detailed information about the communication between these systems, produce an Accounting trace.

For communication between two DB2 subsystems, the Accounting trace provides you with information about the requester locations (for DBATs) and the server locations (for requester threads).

For communication between a DB2 subsystem and a non-DB2 for z/OS requester, OMEGAMON XE for DB2 PE can only present performance data on DBATs. However, the Accounting trace helps you to identify the requester. It supplies the identifier and release level of the requester involved. For requesters from DB2 on multiplatforms it also provides the client platform, application name, authorization ID, and a user-supplied part. If possible, OMEGAMON XE for DB2 PE uses the first eight bytes of the application name to identify the DBAT. If it is not possible to show
the DBAT application name, the plan name is shown as DISTSERV. When
DISTSERV is shown, you can use the MAINPACK identifier to distinguish DBATs.

Accounting trace records are affected by the DB2 subsystem parameter
ACCUMACC, which controls whether and when DB2 Accounting data is
accumulated by the user for DDF and RRSAF threads. A parameter value of 2 or
greater causes Accounting records to roll up into a single record every \( n \)
occurrences of the user on the thread. These values can be set by DDF threads by
Server Connect and Set Client calls, and by RRSAF threads by RRSAF SIGN,
AUTH SIGNON, and CONTEXT SIGNON functions. When roll up occurs, the
values of some fields shown in Accounting reports and traces lose their meanings
because of the accumulation. Thus, these fields are marked as N/P or N/C for
derived fields. For a list of affected fields, see "Fields affected by roll up during
distributed and parallel tasks" on page 235.

Query parallelism considerations
The Accounting report set supports parallel tasks from CP parallelism and Sysplex
query parallelism.

If a query exploits query CP parallelism or Sysplex query parallelism, several tasks
(called parallel tasks) perform the work. For each of these tasks an Accounting
record is generated, which contains counters and timers pertinent to the work
performed by the particular task. In addition, an Accounting record is created that
contains the details about nonparallel work within the thread and data related to
parallel work.

OMEGAMON XE for DB2 PE summarizes all Accounting records that are generated
for such a query and presents the records as one logical Accounting record.
Table 12 describes which values are taken from both the originating and parallel
records and which are taken from the originating record only.

Table 12. Data related to query CP and Sysplex query parallelism

<table>
<thead>
<tr>
<th>Accounting data</th>
<th>Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifiers (such as PRIMAUTH or PLANNAME)</td>
<td>Originating record</td>
</tr>
<tr>
<td>Class 1 elapsed times</td>
<td>Originating record</td>
</tr>
<tr>
<td>Class 1 TCB times</td>
<td>Originating and parallel records</td>
</tr>
<tr>
<td>Class 2 elapsed times</td>
<td>Originating record</td>
</tr>
<tr>
<td>Class 2 TCB times</td>
<td>Originating and parallel records</td>
</tr>
<tr>
<td>Class 7 elapsed times</td>
<td>Originating record</td>
</tr>
<tr>
<td>Class 7 TCB times</td>
<td>Originating and parallel records</td>
</tr>
<tr>
<td>Class 2 and 7 DB2 entry/exit events</td>
<td>Originating record</td>
</tr>
<tr>
<td>Class 3 and 8 times and events</td>
<td>Originating and parallel records</td>
</tr>
<tr>
<td>Class 5 times</td>
<td>Originating record</td>
</tr>
<tr>
<td>SQL counters</td>
<td>Originating record</td>
</tr>
<tr>
<td>RID list counters</td>
<td>Originating and parallel records</td>
</tr>
<tr>
<td>Query parallelism counters</td>
<td>Originating record</td>
</tr>
<tr>
<td>Locking (including data sharing) counters</td>
<td>Originating and parallel records</td>
</tr>
<tr>
<td>RLF data</td>
<td>Originating record</td>
</tr>
<tr>
<td>Buffer pool counters</td>
<td>Originating and parallel records</td>
</tr>
</tbody>
</table>
Table 12. Data related to query CP and Sysplex query parallelism (continued)

<table>
<thead>
<tr>
<th>Accounting data</th>
<th>Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group buffer pool counters</td>
<td>Originating and parallel records</td>
</tr>
<tr>
<td>DDF counters</td>
<td>Originating record</td>
</tr>
<tr>
<td>Data capture counters</td>
<td>Originating record</td>
</tr>
<tr>
<td>SU counters</td>
<td>Originating and parallel records</td>
</tr>
</tbody>
</table>

The elapsed time is taken from the originating record, while CPU and suspension times are calculated from all parallel and originating records. Consequently, both CPU time and suspension times can be larger than the elapsed time. Therefore, you can only get the full picture of response time distribution if the times for each participating task are known. If you suspect that the CPU times or suspension times for a thread where query CP or Sysplex query parallelism is used are, for other reasons, larger than the times being added for several tasks, produce a long Record trace for IFCID 3. This way all parallel and originating Accounting records are reported separately.

In Sysplex query parallelism, the CPU times of the parallel records are normalized so that you can add up the times across multiple DB2s running on different machines. Normalized means that the CPU times are converted to a common unit, called *service unit (SU)*, using a conversion factor. The conversion factor depends on the machine being used.

By default, Accounting trace records from parallel query tasks are rolled up into the originating task's Accounting trace. This is controlled by the DB2 subsystem parameter PTASKROL (parallel tasks roll up), which has a default value of YES. A roll up record is written when the parent task (agent) deallocates on an originating DB2, or when an accumulating child task is deallocated on an assisting DB2. The rolled up data is an accumulation of all counters for that field for each child task that completed and deallocated. When roll up occurs, the values of some fields in Accounting reports and traces lose their meanings because of the accumulation. Thus, these fields are marked as N/P, or N/C for derived fields. For a list of affected fields, see "Fields affected by roll up during distributed and parallel tasks" on page 235.

**Timing information**

The Accounting report set provides a multitude of timing information for nondistributed and distributed transactions.

OMEGAMON XE for DB2 PE reports the following timing information:

- Application times (class 1)
- DB2 times (class 2)
- Suspension times (class 3) and counts
- IFI times (class 5)
- DB2 times on a per package/DBRM basis (class 7)
- Suspensions times on a per package/DBRM basis (class 8) and counts

If an Accounting record represents an allied-distributed thread, additional time fields for DDF data are reported. These DDF times are obtained from Accounting class 1.

The following figure shows an example of the DB2 Accounting times for a nondistributed transaction that does not exploit CP parallelism and how the Accounting times relate to each other and to the DB2 SQL call activity. The
example depicts an application for IMS or CICS. Classes 5, 7, 8, and 9 are not shown. Class 5 is a subset of class 2 and is present only if IFI processing takes place. Classes 7 and 8 are equivalent to classes 2 and 3, but on a package or DBRM level.

The following figure shows various elapsed times for a distributed transaction originating from TSO allied space. CP parallelism is not exploited. The times for both the requester and the server are shown. This figure is a simplified presentation of the processes at the participating locations. It does not, for example, show block fetch statements and it is only applicable to a single row retrieval. DDF-at-Server time is not provided if the transaction uses DRDA.

Figure 85. DB2 Accounting times for a nondistributed transaction

The following figure shows various elapsed times for a distributed transaction originating from TSO allied space. CP parallelism is not exploited. The times for both the requester and the server are shown. This figure is a simplified presentation of the processes at the participating locations. It does not, for example, show block fetch statements and it is only applicable to a single row retrieval. DDF-at-Server time is not provided if the transaction uses DRDA.
The various elapsed times in the figure are defined as follows:

- **DDF at Server**
  
  This is the ELAPSED SER field that is reported in the Requester Fields for Server Location section of the Accounting reports and traces. It represents the elapsed time spent at the server between the time the SQL statement is received and the time the answer is sent to VTAM®. This time is not applicable to DRDA and N/A is printed.

- **DDF at Requester**
This is the ELAPSED REQ field in the Server Fields for Requester Location section of the Accounting reports and traces. It represents the elapsed time spent at the requester between the time the SQL statement is sent and the time the answer from the server is received.

- **Class 1 at Server Location**
  This is the ELAPSED TIME field under the APPLICATION TIMES (CLASS 1) column of the Accounting reports and traces for the server location. It represents the class 1 elapsed time from creation to termination of the DBAT.

- **Class 2 at Server Location**
  This is the ELAPSED TIME field under the DB2 TIMES (CLASS 2) column of the Accounting reports and traces for the server location. It represents the elapsed time to process the SQL requests and the commits at the server.

- **Class 3 at Server Location**
  This is the ELAPSED TIME for the total of the class 3 suspensions in the Accounting reports and traces for the server location. It represents the time the DBAT was suspended while waiting for various system events such as locking, I/O, and other requests.

- **Class 3 at Requester Location**
  This is the ELAPSED TIME column for class 3 suspensions on the Accounting reports and traces for the requester location. It represents the time the allied-distributed thread was suspended while waiting for various system events such as locking, I/O, and other requests.

- **Class 2 at Requester Location**
  This is the ELAPSED TIME field under the DB2 TIMES (CLASS 2) column of the Accounting reports and traces for the requester location. It represents the elapsed time between the application passing the SQL statement to the local (requester) DB2 system and back again. This is the time spent in DB2.

- **Class 1 at Requester Location**
  This is the ELAPSED TIME field under the APPLICATION TIMES (CLASS 1) column of the Accounting reports and traces for the requester location. It represents the elapsed time from creation to termination of the allied-distributed thread.

**Input for Accounting reports**

The Accounting report set can process data originating at different DB2 locations in the same OMEGAMON XE for DB2 PE run.

Several input data sets (in any SMF, GTF, DPMOUT, and Online Monitor trace output data set format) can be logically concatenated in the DD statements for INPUTDD. The data is sorted in the primary sequence of location and reported according to the parameters specified in the Accounting command.

**DB2 traces used in Accounting:**

The Accounting report set requires the following DB2 trace classes and DB2 trace types to be active.

Accounting uses the DB2 IFCIDs 3 and 239 (DB2 trace type Accounting, DB2 trace class 1) as input for the reports and traces. Timing data is taken from all classes:

- Application times (class 1)
- DB2 times (class 2)
- Suspension times (class 3) and counts
- IFI times (class 5)
- DB2 times on a per package/DBRM basis (class 7)
• Suspensions times on a per package/DBRM basis (class 8) and counts

If an Accounting record represents an allied-distributed thread, additional time fields for DDF data are reported. These DDF times are obtained from Accounting class 1.

Accounting trace classes are started either at DB2 start time, by using the installation panel DSNTIPN, or by entering the DB2 START TRACE command at the terminal. For example:

```
-START TRACE(ACCTG) CLASS(1,2,3,5,7,8) DEST(SMF)
```

The Accounting classes 1 and 3 should always be active. The overhead is not significant, whereas the information they provide is crucial for all aspects of performance monitoring. It is also useful to always have class 2 active. If you do not always have class 3 active, activate class 3 when you activate class 2.

Classes 7 and 8 provide valuable information on a per package or per DBRM basis. They are equivalent to classes 2 and 3. Package or DBRM Accounting is probably most beneficial for a DB2 server of a non-DB2 requester or when an application plan can execute many packages or DBRMs, but actually executes 10 or less for a given Accounting record. If class 2 is active, also activate class 7 if package Accounting is needed. If class 3 is active, also activate class 8 if package Accounting is needed. More than 10 packages or DBRMs introduce additional overhead in data collection reducing the benefit versus cost. However, this cost is still significantly less than a DB2 performance class trace.

**Accounting record generation:**

About how and when Accounting data is collected and recorded.

The collection of Accounting data begins when a thread connects to DB2. A completed Accounting record is written when:
• The thread terminates
• The authorization identifier changes because the thread is reused
• A DBAT becomes inactive
• A parallel task completes when query CP or Sysplex query parallelism is exploited

**Note:**

• NEW USER is reported when DB2 Accounting records are reported in the IMS thread and CICS thread reuse situations where a new AUTHID is used.
• If TOKENE=YES on the TYPE=ENTRY statement in the RCT table is specified, the CICS attachment facility requests an Accounting record to be produced during thread reuse even if the user authorization ID does not change. A CICS LU6.2 token is also passed to DB2 allowing correlating CICS and DB2 trace records.
• For thread reuse, all values are accumulated since the last Accounting record.
• When a CICS application program causes more than one SYNC point commit or rollback, DB2 can produce several out-of-sequence Accounting records for the application if CICS attach thread swapping occurs.
• If a thread is reused with the same user authorization ID and TOKENE has not been specified, the DB2 Accounting record represents several CICS transactions.
Missing data sections:

Conditions that might cause missing data for Accounting reports.

The following DB2 threads might result in missing data sections because required data could not be gathered:

- If attaching to QMF that generates a unique DB2 thread.
  Under this thread no SQL, buffer, or locking information is gathered.
- If running a dynamic SQL that generates a unique DB2 thread.
  Under this thread, SQL, buffer, and locking activity can be performed. A data section is produced if activity in the data section occurs. Resource limit activity is only reported in the Accounting record for dynamic SQL activity.
- If ending a QMF session that creates a unique DB2 thread.
  Under this thread no SQL, buffer, or locking information is gathered.
- If running a batch job that creates a unique DB2 thread.
  Under this thread SQL, buffer, and locking activity can be performed. No resource limit activity is tracked for this thread.
- If location A uses DRDA protocol to access data at location B and does not do any local work, no SQL information is gathered for the corresponding allied-distributed thread at location A.

Creating effective Accounting reports

For a report to be effective it must deliver the information you need efficiently in terms of resources required to produce it, the volume of data produced, and the time it takes for you to interpret the report.

The following topics provide additional information:

- "Processing considerations"
- "Exception processing" on page 133
- "ORDER processing" on page 135
- "Example of ordering by interval" on page 142
- "TOP processing" on page 143
- "Reducing data" on page 146
- "Processing intervals" on page 147
- "Member-scope and group-scope reporting" on page 157

Processing considerations

Because Accounting is one of the most frequently used report sets, consider the effects on batch resources. To reduce processing time when producing Accounting reports and traces, only ask for the information that you really need.

Before producing a report or trace, consider how much detail you need.

Use the short (default) layouts of reports and traces whenever possible. In most instances these provide enough detail for monitoring and problem determination.

A detailed report that uses all input data that was gathered for a long period uses a lot of system resources. The result is pages of information that you are probably not interested in.

To avoid unnecessary processing overhead and to save time:
Choosing the right level of detail:

Before you produce a report or trace, consider how much detail you need. Use the LAYOUT subcommand option of the Accounting reports and traces to control the amount of data to be produced.

In most situations the default layouts, which are short versions of reports and traces, provide enough detail for monitoring and problem determination. Do not use the most comprehensive layouts that show all possible fields unless you need this information.

If the use of commands, subcommands, and subcommand options does not provide sufficient control over the contents of reports and traces, you can use the User-Tailored Reporting (UTR) feature to create and tailor your own report and trace layouts. With UTR you can add, remove, and change individual fields and entire report blocks to control the volume, contents, and layouts of your reports and traces. For more information, see "Tailoring report layouts" on page 198.

Filtering data:

You can limit the amount of data to be processed by filtering the input data. You can specify filters in the GLOBAL command and in the REDUCE, REPORT, TRACE, and FILE subcommands.

Specify the filters in GLOBAL whenever you can, because only the data that passes through the GLOBAL filters is processed further. The less data OMEGAMON XE for DB2 PE needs to process, the better the performance.

However, ensure that you do not exclude records that are needed in OMEGAMON XE for DB2 PE processing.

FROM and TO subcommand options:

The simplest filter is the start and end date and time of the data to be reported. Specify the start and end dates and times by using the FROM and TO subcommand options.

For example, to monitor the performance of your system only during peak hours and to produce the default version of a report, specify:

```
GLOBAL
   FROM (09/25/08,08:30)
   TO (09/25/08,17:00)
```
ACCOUNTING

The report shows information from 8:30 a.m. to 5:00 p.m. for the specified day.

Presuming that the input data set contains data for more than one day, for example a week, you can generate a report that shows the performance of your system during peak hours for the whole week by specifying:

GLOBAL
  FROM (,08:30)
  TO (,17:00)
ACCOUNTING

The report shows information from 8:30 a.m. to 5:00 p.m. for every day of the week.

INCLUDE and EXCLUDE subcommand options:

Another way to filter data is to include data only for particular OMEGAMON XE for DB2 PE identifier values, for example, user IDs or plans. You can do this by using the INCLUDE and EXCLUDE subcommand options.

For example, if you have a problem with applications coming from one location, USIBMSNEWY11, and you know the CICS transactions are not causing a problem, specify:

GLOBAL
  INCLUDE (LOCATION (USIBMSNEWY11))
  EXCLUDE (CONNTYPE (CICS))

Continuing the example, suppose the report indicated a problem with authorization identifier USERID01 using plan PVLDD4C3. You are only interested in data belonging to that user ID and plan. You can now specify:

GLOBAL
  INCLUDE (LOCATION (USIBMSNEWY11))
  INCLUDE (AUTHID (USERID01))
  INCLUDE (PLANNAME (PVLDD4C3))

Suppressing the OMEGAMON XE for DB2 PE internal sort:

When you request Accounting functions only, it is often possible to avoid the OMEGAMON XE for DB2 PE internal sort of the input data. Suppressing the internal sort with the PRESORTED option of the GLOBAL command reduces the size of the sort work files that must be allocated and the processing time.

The PRESORTED option of the GLOBAL command controls the internal sort. For example, to produce an Accounting report without sorting the input data, specify:

GLOBAL
  PRESORTED (ACCEPT)
ACCOUNTING

The resulting Accounting report shows Accounting data for all locations in the input data set, without performing an internal sort.

**Grouping data by defining sets of OMEGAMON XE for DB2 PE identifiers:**

Use the GROUP command to define a set of OMEGAMON XE for DB2 PE identifier values that can be used when you request certain reports. The information for the set is reported as a single entry in the reports.

In GROUP processing the data for all items of the set is consolidated into one record. This improves OMEGAMON XE for DB2 PE performance because fewer records need to be processed.

Sets are also useful for reporting purposes such as to report data for an entire department instead of every individual person.

This is how you specify sets. Assume that your sales department consists of three users, USER001, USER002, and USER003. You want to produce an Accounting report with performance data for that department. You can specify:

```
GROUP (
  PRIMAUTH (
    SALES (USER001, USER002, USER003)))
```

The Accounting report shows information for the sales department as a single entry.

The OMEGAMON XE for DB2 PE identifiers where GROUP can be applied in Accounting are:
- CONNECT
- CONNTYPE
- CORRNAME
- CORRNMBR
- MAINPACK
- ORIGAUTH
- PLANNAME
- PRIMAUTH
- REQLOC

**Specifying intervals for long-term reporting:**

If you want to perform a trend analysis, you probably want to report data by interval. To do this, first reduce the input data for your Accounting reports using an interval other than the default zero. Then, order the report by intervals.

For example, you might want to report the data at daily intervals. In this case you would specify INTERVAL(1440) in the REDUCE subcommand and
ORDER(INTERVAL) in the REPORT subcommand. If you do not intend to produce reports at time intervals, use the default (0) for INTERVAL, which means that no interval processing is performed.

**Specifying exception thresholds for specific fields:**

You can set exception thresholds for virtually all Accounting fields. However, carefully consider the fields for which to specify exception thresholds. The more fields you specify, the greater the effects on processing.

Specify exception thresholds only for those fields that you believe will signal poor performance in your environment.

You can also use the TOP (ONLY) subcommand option to identify problems instead of specifying exception thresholds. TOP processing is more economical than exception processing.

**Using a DPMOUT data set:**

Keep the content of the DPMOUT data set for further reports.

After OMEGAMON XE for DB2 PE has generated all reports that you requested, the preprocessed input data is either discarded or written to the output data set DPMOUT, depending on what you specified. The DPMOUT data set can be used as input to OMEGAMON XE for DB2 PE, so if you want to produce reports from the same data in several executions and your SMF/GTF data set is large, you might want to keep the DPMOUT data set.

If you do not specify DPMOUTDD, only the records required for the current job step are processed, which improves performance. If you want to keep a DPMOUT data set for the specific purpose of producing future Accounting reports and traces, specify GLOBAL INCLUDE IFCID(3 239) in the command stream within the JCL. This reduces the size of the DPMOUT data set.

**Saving reduced data:**

Use the REDUCE and SAVE subcommand options to store historical Accounting data in smaller data sets.

If you need to keep historical Accounting data about DB2 performance, consider using REDUCE and SAVE. Reducing and saving data uses a considerable amount of system resources, but the resulting Save data set is much smaller than the original input data set.

The size of the data set depends on the reduction interval that you specified and the type of environment the data is from (for example, the number of different users and plans present in the input data), but it is always much smaller than the original input data set.

You can produce reports from the reduced and saved data by using the RESTORE command. Remember that you cannot produce traces from reduced data.

**Exception processing**

Use exception processing to identify Accounting report, trace, and File data set entries with fields that contain values outside thresholds specified in the Exception Threshold data set. When exception processing is active, the data to be reported is
checked against these thresholds. You can obtain exception reports, traces, and File data sets by using the EXCEPTION subcommand option in ACCOUNTING subcommands.

There are two threshold types that you can set: warning and problem. A warning message is printed if a value is outside the first threshold. A problem message is printed if a value is outside the second threshold.

Accounting exception reports, traces, and File data sets are identical to the usual Accounting reports, traces, and File data sets, except that they only contain entries that have at least one field in exception status. For reports and traces, a block of data that shows the fields in exception status is printed next to the report or trace entry.

Exception logs contain Accounting and Statistics fields that are in exception status in a single report in timestamp order.

The Exception Log data set is written if you define the EXTRCDD1 DD statement in the JCL. To prevent the generation of the Exception Log data set, omit the EXTRCDD1 statement from your JCL (the preferred method), or specify DUMMY in the definition.

The Exception Log File data set is written if you define the EXFILDD1 DD statement in the JCL. To prevent the generation of the Exception Log File data set, omit the EXFILDD1 statement from your JCL (the preferred method), or specify DUMMY in the definition.

As the following figure shows, if EXCEPTION was specified in the REPORT or TRACE subcommand, the entry is formatted and printed in the requested layout, followed by the Exception Messages block. The Exception Messages block identifies the fields in exception status.

```plaintext
<table>
<thead>
<tr>
<th>TYPE</th>
<th>FIELD ID</th>
<th>FIELD DESCRIPTION</th>
<th>BY</th>
<th>VALUE</th>
<th>THRESHOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROBLEM</td>
<td>QXINCRB</td>
<td>INCREMENTAL BINDS</td>
<td>TOTAL</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>PROBLEM</td>
<td>ASCDML</td>
<td>TOTAL SQL DML STATEMENTS</td>
<td>COMMIT</td>
<td>127.25</td>
<td>&gt; 100</td>
</tr>
<tr>
<td>WARNING</td>
<td>QBACGET</td>
<td>GETPAGES</td>
<td>THREAD</td>
<td>2589.00</td>
<td>&gt; 2500</td>
</tr>
</tbody>
</table>
```

Figure 87. Accounting Exception Messages block

The following columns are presented in the Exception Messages block:

- **TYPE** The type of exception. Valid values are WARNING and PROBLEM.
- **FIELD ID** The name of the field from the Exception Threshold data set.
- **FIELD DESCRIPTION** The description of the field.
- **FIELD QUALIFIER** The qualifier of the field. It can be blank (for the majority of fields), a buffer pool ID, group buffer pool ID, package or DBRM name, or a remote location name.
- **BY** The basis for the comparison. The following information can be printed in this column:
TOTAL
The threshold is specified as a “total” value.

MINUTE
The threshold is specified as a “by minute” value. The value in the report or trace entry is divided by the number of minutes (class 1 elapsed time) before making the comparison.

SECOND
The threshold is specified as a “by second” value. The value in the report or trace entry is divided by the number of seconds (class 1 elapsed time) before making the comparison.

COMMUT
The threshold is specified as a “by commit” value. The value in the report or trace entry is divided by the number of Commits before making the comparison.

THREAD
The threshold is specified as a “by thread” value. The value in the report or trace entry is divided by the number of threads before the comparison. For traces it is equivalent to by total.

VALUE
The actual field value that is used for the comparison. For “by minute”, “by second”, “by commit”, or “by thread” comparisons, the value that you get after the division is printed. The greater than symbol (> or the less than symbol (<) is printed between this column and the THRESHOLD column to indicate whether the value is larger or smaller than the threshold value.

THRESHOLD
The threshold defined in the Exception Threshold data set.

ORDER processing
Unless the TOP (ONLY) subcommand option was specified, data in Accounting traces is presented in a chronological order and data in Accounting reports is summarized by plan names within primary authorization IDs. You can change the way reports are summarized by using the ORDER subcommand option of the REPORT subcommand.

Use ORDER to specify the OMEGAMON XE for DB2 PE identifiers that are to be used in aggregating Accounting records. Unless the TOP (ONLY) subcommand option was specified, ORDER also determines the sequence for sorting a report. You can aggregate records by using one, two, or three identifiers, separated by a dash, and specifying up to five sets of these identifiers per entry of ORDER. You can specify one entry of ORDER per REPORT subcommand.

The default for ORDER is PRIMAUTH-PLANNAME.

In addition to these OMEGAMON XE for DB2 PE identifiers, you can use the REDUCE INTERVAL to order data on Accounting and Statistics reports. See "Example of ordering by interval" on page 142 for an example of a report ordered by interval.

The following examples demonstrate ORDER processing.

Examples of using ORDER:
These examples show the same input data ordered in three different ways in an Accounting report.

**Default ordering**

The following figure shows a report with default ordering of plan names within primary authorization IDs.

<table>
<thead>
<tr>
<th>PRIMAUTH</th>
<th>PLANNAME</th>
<th>#OCCURS</th>
<th>#ROLLBK</th>
<th>SELECTS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>CLASS1</th>
<th>EL.TIME</th>
<th>CLASS2</th>
<th>EL.TIME</th>
<th>GETPAGES</th>
<th>SYN.READ</th>
<th>LOCK</th>
<th>SUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMF001</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12:22:50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DDL03P04</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/P</td>
<td>30.00</td>
<td>N/P</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNUTIL</td>
<td>0</td>
<td>87</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/P</td>
<td>157.00</td>
<td>N/P</td>
<td>1.50</td>
<td>121.00</td>
<td>3.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TOTAL

<table>
<thead>
<tr>
<th>PRIMAUTH</th>
<th>PLANNAME</th>
<th>#OCCURS</th>
<th>#ROLLBK</th>
<th>SELECTS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>CLASS1</th>
<th>EL.TIME</th>
<th>CLASS2</th>
<th>EL.TIME</th>
<th>GETPAGES</th>
<th>SYN.READ</th>
<th>LOCK</th>
<th>SUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMF001</td>
<td></td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3:02:00</td>
<td>0.00</td>
<td>N/P</td>
<td>298.77</td>
<td>9.77</td>
<td>1.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL03P04</td>
<td></td>
<td>103</td>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
<td>0.95159</td>
<td>N/P</td>
<td>143.23</td>
<td>3.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TOTAL

<table>
<thead>
<tr>
<th>PRIMAUTH</th>
<th>PLANNAME</th>
<th>#OCCURS</th>
<th>#ROLLBK</th>
<th>SELECTS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>CLASS1</th>
<th>EL.TIME</th>
<th>CLASS2</th>
<th>EL.TIME</th>
<th>GETPAGES</th>
<th>SYN.READ</th>
<th>LOCK</th>
<th>SUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMF001</td>
<td></td>
<td>103</td>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
<td>0.95159</td>
<td>N/P</td>
<td>143.23</td>
<td>3.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL03P04</td>
<td></td>
<td>103</td>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
<td>0.95159</td>
<td>N/P</td>
<td>143.23</td>
<td>3.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### GRAND TOTAL

The data is ordered according to various authorization IDs and plans. All primary authorization IDs and plans present in the input data are shown.

The TOTAL rows are printed for primary authorization IDs that contain more than one plan.

**Ordering by connection ID**

You can order the report by the connection ID as follows:

```
ACCOUNTING
REPORT
```
ORDER (CONNECT)

The following figure shows a report with ordering by connection ID.

<table>
<thead>
<tr>
<th>CONNECT</th>
<th>OCCURS</th>
<th>DISTR</th>
<th>SELECTS</th>
<th>UPDATES</th>
<th>DEL SELECTS</th>
<th>EL. TIME</th>
<th>BUFFER UPDATES</th>
<th>TOT. PREF</th>
<th>LOCKOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCH</td>
<td>7</td>
<td>200</td>
<td>0.00</td>
<td>0.00</td>
<td>23.14</td>
<td>0.29</td>
<td>1:09:01:529998</td>
<td>N/P</td>
<td>1253.14</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>22</td>
<td>59.43</td>
<td>0.86</td>
<td>0.57</td>
<td>0.43</td>
<td>4.698535</td>
<td>N/P</td>
<td>126.71</td>
</tr>
<tr>
<td>DB2CALL</td>
<td>10</td>
<td>10</td>
<td>0.00</td>
<td>0.00</td>
<td>1:06.989512</td>
<td>N/P</td>
<td>23.00</td>
<td>0.20</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.041995</td>
<td>N/P</td>
<td>0.00</td>
</tr>
<tr>
<td>UTILITY</td>
<td>18</td>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>2:18.108687</td>
<td>N/P</td>
<td>371.67</td>
<td>11.67</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>139</td>
<td>0.00</td>
<td>0.00</td>
<td>1.144894</td>
<td>N/P</td>
<td>181.22</td>
<td>5.33</td>
<td>0</td>
</tr>
</tbody>
</table>

*** GRAND TOTAL ***

<table>
<thead>
<tr>
<th></th>
<th>OCCURS</th>
<th>DISTR</th>
<th>SELECTS</th>
<th>UPDATES</th>
<th>DEL SELECTS</th>
<th>EL. TIME</th>
<th>BUFFER UPDATES</th>
<th>TOT. PREF</th>
<th>LOCKOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>35</td>
<td>212</td>
<td>1.68</td>
<td>0.00</td>
<td>4.63</td>
<td>0.06</td>
<td>15:10.472383</td>
<td>N/P</td>
<td>448.34</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>161</td>
<td>12.17</td>
<td>0.46</td>
<td>0.11</td>
<td>0.37</td>
<td>1:546508</td>
<td>N/P</td>
<td>118.54</td>
</tr>
</tbody>
</table>

Accounting REPORT COMPLETE

Figure 89. Accounting report - ordered by connection ID

Three connection IDs are reported: information for connection ID BATCH is reported on the first two lines followed by information for connections DB2CALL and UTILITY.

GRAND TOTAL is printed at the end of the report. It shows the aggregated values for all three connection IDs.

Ordering by correlation ID

You can also identify the task by using correlation data:

ACCOUNTING

REPORT
ORDER (CORRNAME)

The following figure shows a report with ordering by correlation ID.
Note: In a distributed environment reports should be ordered by REQLOC or CONNTYPE. If REQLOC or CONNTYPE are not used in the ORDER subcommand option of REPORT, the Accounting portion of all threads (including DBATs) where the combination of OMEGAMON XE for DB2 PE identifiers is the same is reported as one entry.

Examples of ordering by Plan, Main Package, and Package:

These examples show data ordered by PLANNAME, MAINPACK, and PACKAGE identifiers.

The data shown in Figure 91 on page 139 is used to produce the reports shown in Figure 92 on page 140, Figure 93 on page 141, and Figure 94 on page 142. The data in Figure 91 on page 139 has been simplified for the purpose of the examples to follow.
The input data contains information about:

- Two plans. PLANX is used by three threads and PLANY by one thread. The first instance of PLANX shows that three packages (PACKA, PACKB, and PACKC) are executed. The elapsed time for the entire PLANX is 10 seconds and CPU times for the individual packages are 0.01, 0.02, and 0.01 seconds.

- Four packages. Different combinations of packages were executed under the different plans, because a plan does not necessarily invoke the same packages each time it is executed. This can happen when, for example, a number of packages are bound in a single plan in a CICS environment and different packages are executed in different circumstances.

**Ordering by Plan**

This example shows the result of ordering the input data by plan. The following command was used to produce the example shown in Figure 92 on page 140:

```
ACCOUNTING
REPORT
```
The report in Figure 92 contains an entry for both plans in the input data.

Data for the different packages is summarized under the plans. Also, different DB2 executions of PLANX are summarized in one entry.

### Ordering by Plan and MAINPACK

The MAINPACK identifier is used to distinguish between records with the same plan name, but which executed different packages.

This example shows the result of ordering the input data by plan and MAINPACK.

MAINPACK identifies a representative package within the plan. The first package ID is the default for MAINPACK. However, in this example, the MAINPACK member of the DPMPARMS data set has been modified so that it is the package ID of the last executed package, see "Defining the MAINPACK identifier" on page 215 for information about MAINPACK.

In the input data shown in Figure 91 on page 139, it is assumed that PACKC is the last executed package in the first entry for PLANX, PACKD for the second and third entries of PLANX, and PACKA for the entry of PLANY.

The following command was used to produce the report in Figure 93 on page 141:

```
ACCOUNTING
REPORT
```
The report in Figure 93 shows an entry for each combination of PLANNNAME and MAINPACK.

There are two entries for PLANX:
- One entry where PACKC is the last package executed.
- Another entry where PACKD is the last package executed. Threads 2 and 4 are combined in this entry.

There is one entry for PLANY with the representative package PACKA.

You cannot use this report to attribute the General Accounting data to one package, unless only one package exists within a plan.

### Ordering by Package or DBRM

The previous examples present packages within plans. To summarize the package Accounting data regardless of the plan under which the packages or DBRMs were executed, you can order by package.

This example shows the result of ordering the input data by package.

The following command was used to produce the report in Figure 94 on page 142:  

```
ACCOUNTING REPORT
```
The report in Figure 94 shows the use of resources on a per package/DBRM basis, regardless of the plan under which a particular package is executed.

Note: Accounting reports that are ordered by package identifier (created by using the PACKAGE keyword with the ORDER subcommand option) show only the following report blocks:
- Package Identification
- Times - Class 7 - Package Times
- Package Suspensions
- Global Contention L-locks
- Global Contention P-locks
- Package Buffer Pool Activity

Example of ordering by interval:

This example shows a short Accounting report that is ordered by interval.

To produce a report that presents DB2 activity by time intervals, the input data first needs to be reduced to the intervals that you want to use in reporting.

The following command was used to generate the sample report in Figure 95 on page 143:

```
ACCOUNTING
REDUCE
   INTERVAL (5)
REPORT
   ORDER (INTERVAL)
```

In this command the data is reduced to 5-minute intervals. The report is ordered by this interval. An entry is produced that shows the activity during every 5-minute interval.
Reports ordered by INTERVAL, with or without another identifier, are especially useful in trend analysis. For example, you can reduce and save your data by specifying INTERVAL(1440) to produce reports that show the day-by-day activity of your DB2 subsystem.

**TOP processing**

TOP processing is useful to identify the report entries that might indicate a problem application.

Signs of a problem can be a long elapsed or processing time, a high number of suspensions, commits per update, or buffer updates. Identifying these entries is especially useful when your report covers many users or a long period so that it is not immediately clear which applications are causing performance problems.

To identify report entries with a high value in certain fields, you can produce an Accounting report or trace with TOP lists. TOP lists are index-like reports at the end of a report or trace. They point out the most interesting entries in the report or trace. You can generate such lists by using the TOP subcommand option.

**Note:** Entries with 0 or undetermined values are not shown. Moreover, if your input data contains only 0 or undetermined values for the TOP fields requested, a TOP list is not generated.

You can also filter a report or trace so that only the main resource consumers are shown, ordered by descending resource value. To obtain a filtered report or trace, specify the TOP subcommand option with the ONLY keyword.

The shown resource values are average values. To obtain total values, which means, the main resource consumers calculated by taking into account how often they consumed resources, specify the TOP subcommand option with the TOTAL keyword. TOTAL only applies to reports. If you specify it with a trace, it is ignored.

**Examples of TOP processing:**

These examples show some applications of the TOP subcommand option in Accounting reports and traces.
Top-10 plans in Accounting report

To produce a short Accounting report with a list of the top-10 plans that spent the longest time in an application, specify:

```
GLOBAL
   INCLUDE (LOCATION (USIBMSNEWY11))
   EXCLUDE (CONNTYPE (CICS))
ACCOUNTING
   REPORT
      ORDER (PLANNAME)
      TOP
```

The last page of the report shows a list of the 10 plans that had the highest value for elapsed time in application, as shown in the following figure. The TOP list also shows the page on which the report entries can be found.

```
PLANNAME       VALUE   PAGE
1 PVLDD4C3     13:36.924222 1-1
2 PVLDD4B3     11:25.333551 1-1
3 PVLDD4A1     10:19.155585 1-1
4 PVLDD4A4     9:18.056786  1-1
5 DSNB500      8:50.415594  1-1
6 DSNB501      8:49.390683  1-1
7 PVLDD4B6     7:23.338584  1-1
8 PVLDD4A6     4:26.046783  1-1
9 PVLDD4C4     3:10.939686  1-1
10 PVLDD402    2:34.389670  1-1
```

Figure 96. Accounting report - TOP listing

Top-5 Getpages in Accounting trace

To produce a trace with a TOP list for the number of Getpage requests, use the following command:

```
ACCOUNTING
   TRACE
      TOP (5 GETPAGES)
```

The TOP list on the last page of the trace consists of the five entries that had the highest number of Getpage requests, as shown in Figure 97 on page 145. Each entry consists of TIMESTAMP, PRIMAUTH, and PLANNAME information, the value of the number of Getpage requests, and the page number where the trace entry can be found.
TOP lists of available fields in Accounting report

You can generate Accounting reports and traces that show TOP lists for all fields available for use with the TOP subcommand option, as shown in the following command:

```
...
ACCOUNTING
   REPORT
      TOP (*)
...
```

Top-3 default field in Accounting trace

To produce a filtered trace that shows only the top-3 entries for the default TOP field, namely ELAPSED TIME IN APPLICATION, use the following command:

```
...
Accounting
   TRACE
      TOP (3 ONLY)
...
```

Figure 98 on page 146 shows what such a trace looks like.
Note: The TOP(ONLY) subcommand option changes the presentation sequence of Accounting reports and traces. Report and trace entries are ordered according to the TOP resource instead of the timestamp or ORDER subcommand options. The summarization in reports is not affected by the changed sequence caused by TOP filtering.

Reducing data
Use the REDUCE subcommand to consolidate records with certain common characteristics into a single record.

You can limit the range of records by date and time. You can specify multiple ranges of time. This can be useful for monitoring peek-time performance, for example.

Within a specified time range, you can specify the interval at which records are consolidated.

The start time of the first interval that is processed by REDUCE is influenced by BOUNDARY, INTERVAL, and FROM.

OMEGAMON XE for DB2 PE attempts to reduce all data that falls between FROM and TO dates and times. The first interval starts at a time aligned with BOUNDARY, at or before the FROM time. If an interval cannot be aligned with the FROM time, the first properly aligned interval starting before the FROM time is used.

Although there is no restriction on the INTERVAL and BOUNDARY combination, your specification should comply with the following rules:

- For intervals of less than 60 (excluding 0), there should be a whole number of intervals in an hour. Choose one of the following values:
  - 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30
- For intervals of 60 or greater, there should be a whole number of intervals in a day. Choose one of the following values:
  - 60, 120, 180, 240, 360, 480, 720, or 1440
- For intervals of one day (1440) or greater, INTERVAL should be a multiple of 1440.
- Select your interval and boundary so that the first interval starts at the FROM time.
Examples of interval calculation:

These examples show how the REDUCE subcommand is used in combination with FROM, TO, INTERVAL, and BOUNDARY to align start times and intervals.

Using REDUCE to align to the start of the hour

```
REduce
  from (,08:00)
  to (,10:00)
  interval (30)
  boundary (60)
```

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the FROM time (08:00). Subsequent intervals start every 30 minutes (08:30, 09:00, and 09:30 each day).

Using REDUCE to cover a day

```
reDuce
  interval (1440)
  boundary (60)
```

The following defaults are applied:

- The default for FROM is all dates and a time of 00:00:00.00
- The default for TO is all dates and a time of 23:59:59.99

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the FROM time (00:00). Subsequent intervals cover 1 440 minutes or one day. An interval starts at 00:00 each day.

Using REDUCE to start every hour

```
reDuCE
  from (,08:30)
  to (,12:00)
  interval (60)
  boundary (60)
  report
  from (,08:30)
  to (,12:00)
```

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the hour of the FROM time (08:00). Subsequent intervals start every hour (09:00, 10:00, and 11:00).

Processing intervals:

Interval processing within the REDUCE subcommand determines the time intervals that are used for reducing Accounting data and it influences how data is reported.

Use intervals for:

- Reporting by intervals
- Producing reports with different time spans
If SAVE accompanies REDUCE, the reduced data is saved into the Save data set.

This following topics provide examples of interval processing.

*How intervals are calculated:*

The start time of the first interval that is processed by REDUCE is influenced by
BOUNDARY, INTERVAL, and FROM.

OMEGAMON XE for DB2 PE attempts to reduce all data that falls between the
FROM and TO times. The first interval starts at a time aligned with BOUNDARY, at
or before the FROM time. If an interval cannot be aligned with the FROM time, the
first properly aligned interval starting before the FROM time is used. An interval that
starts before the FROM time only contains data between the FROM time and the
start of the next interval. Input data before the FROM time is not processed.

Although there is no restriction on the INTERVAL and BOUNDARY combination,
your specification should comply with the following rules:

- For intervals of less than 60 (excluding 0), there should be a whole number of
  intervals in an hour. Choose one of the following values:
  - 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30
- For intervals of 60 or greater, there should be a whole number of intervals in a
day. Choose one of the following values:
  - 60, 120, 180, 240, 360, 480, 720, or 1440
- For intervals of one day (1440) or greater, INTERVAL should be a multiple of
  1440
- Select your interval and boundary so that the first interval starts at the FROM
time

If you do not require interval processing, do not change the default INTERVAL (0)
for performance reasons.

Always use the largest interval that meets your reporting requirements. For
example, if daily reports provide sufficient granularity, use INTERVAL (1440).

*Examples of interval processing:*

Sample of SMF File data used for the examples.

The following examples assume:

1. The DB2 instrumentation facility is started for Accounting to SMF at DB2 startup
by the following DB2 command:
   ```
   START TRACE (ACCTG) DEST(SMF) CLASS(1)
   ```
2. The system programmer wants to analyze performance for Thursday, 14 March
   1999, so the SMF file for that day is obtained.
   
   The following figure represents sample data from the SMF file. Actual trace data
can have a different distribution of DB2 records and timestamps. For the
purpose of this example, assume that the Statistics trace was previously active.
3. OMEGAMON XE for DB2 PE is run to get Accounting reports.

**Examples using REDUCE:**

These examples show how the start time is aligned by the use of BOUNDARY.

<table>
<thead>
<tr>
<th>DB2 Timestamp</th>
<th>Sequence Destination #</th>
<th>FCID</th>
<th>Record Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/14/08 8:30:00</td>
<td>0001</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 8:30:01</td>
<td>0002</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 8:31:05</td>
<td>0003</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 8:45:13</td>
<td>0004</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 8:57:27</td>
<td>0005</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 8:59:59</td>
<td>0006</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 9:00:00</td>
<td>0007</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 9:12:11</td>
<td>0008</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 9:15:00</td>
<td>0009</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 9:29:59</td>
<td>0010</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 9:30:00</td>
<td>0011</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 9:30:01</td>
<td>0012</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 9:32:00</td>
<td>0013</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 9:43:00</td>
<td>0014</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 9:55:59</td>
<td>0015</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 10:00:00</td>
<td>0016</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 10:00:00</td>
<td>0017</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 10:30:00</td>
<td>0018</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 10:30:01</td>
<td>0019</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 11:00:00</td>
<td>0020</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 11:00:01</td>
<td>0021</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 11:29:59</td>
<td>0022</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 11:30:00</td>
<td>0023</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 12:00:01</td>
<td>0024</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 12:00:01</td>
<td>0025</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 12:15:00</td>
<td>0026</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 12:17:54</td>
<td>0027</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 12:30:00</td>
<td>0028</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 12:30:00</td>
<td>0029</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 13:00:00</td>
<td>0030</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 13:00:01</td>
<td>0031</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 13:10:31</td>
<td>0032</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 13:15:00</td>
<td>0033</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 13:17:54</td>
<td>0034</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 13:23:34</td>
<td>0035</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 13:27:00</td>
<td>0036</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 13:30:00</td>
<td>0037</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 13:30:00</td>
<td>0038</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 13:37:30</td>
<td>0039</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 15:20:00</td>
<td>0040</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 15:20:00</td>
<td>0041</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 15:37:30</td>
<td>0042</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 15:50:00</td>
<td>0043</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 15:50:01</td>
<td>0044</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 16:15:00</td>
<td>0045</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 16:15:00</td>
<td>0046</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 16:28:02</td>
<td>0047</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 16:30:31</td>
<td>0048</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 16:44:00</td>
<td>0049</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 16:44:59</td>
<td>0050</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 16:45:00</td>
<td>0051</td>
<td>0001</td>
<td>System statistics</td>
</tr>
<tr>
<td>7/14/08 16:45:01</td>
<td>0052</td>
<td>0002</td>
<td>Database statistics</td>
</tr>
<tr>
<td>7/14/08 16:51:06</td>
<td>0053</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 16:54:17</td>
<td>0054</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 16:55:37</td>
<td>0055</td>
<td>0003</td>
<td>Accounting</td>
</tr>
<tr>
<td>7/14/08 16:57:59</td>
<td>0056</td>
<td>0003</td>
<td>Accounting</td>
</tr>
</tbody>
</table>

Figure 99. Example SMF File data
Example 1:

ACCOUNTING
REDUCE
  FROM (,09:00)
  TO (,11:00)
  INTERVAL (30)
  BOUNDARY (60)

The data that is available for reducing is limited by the GLOBAL command. If you specified FROM and TO dates and times in GLOBAL, OMEGAMON XE for DB2 PE discards all records outside those dates and times before reducing. In this example no GLOBAL command is in effect, so the FROM and TO dates and times that are specified with REDUCE are used. No dates are specified, so all dates are included.

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the FROM time (09:00). Subsequent intervals start every 30 minutes. The following intervals are calculated:
- Beginning at 9:00:00 - containing two Accounting records
- Beginning at 9:30:00 - containing four Accounting records

No intervals are calculated at 10:00:00 or 10:30:00 because there are no Accounting records.

The interval begin and end times are reported in the job summary report. The job summary report indicates that an interval is calculated and indicates the number of records processed during the interval. It does not indicate the number of consolidated records after reducing.

Example 2:

ACCOUNTING
REDUCE
  INTERVAL (1440)
  BOUNDARY (60)

In this example no GLOBAL command is in effect and no dates or times have been specified in REDUCE, so the following defaults are applied:
- The default for FROM is all dates and a time of 00:00:00.00
- The default for TO is all dates and a time of 23:59:59.99

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the FROM time (00:00). Subsequent intervals cover 1440 minutes or one day. An interval starts at 00:00 each day. In this example, there is one interval that contains all of the Accounting records in the input data set.

The interval begin and end times are reported in the job summary report.

Examples using REDUCE and REPORT:

These examples show the effect of REPORT FROM and TO on REDUCE INTERVAL and BOUNDARY.
Example 1

 ACCOUNTING
 REDUCE
 INTERVAL (60)
 REPORT
 FROM (, 9:00:00)
 TO (,12:00:00)

In this example no GLOBAL command is in effect and no FROM and TO times are specified in the REDUCE subcommand, so all records in the input are reduced.

The REDUCE subcommand specifies that data is reduced at 60-minute intervals. By default, the boundary is set to 60. Data is gathered and consolidated every hour on the hour.

The Accounting data starts at 8:31:05. The INTERVAL begins on the hour. OMEGAMON XE for DB2 PE determines that the record at 8:31:05 falls into an interval beginning at 8:00:00 and sets the beginning interval time to 8:00:00.

Each additional Accounting record is read. The Accounting records at timestamps 8:31:05, 8:45:13, and 8:57:27 all fall into the interval beginning at 8:00:00. These three records are accumulated (added, subtracted, or whatever is appropriate for the given Accounting fields), consolidated, and stored by unique OMEGAMON XE for DB2 PE identifier sets (PRIMAUTH, PLAN, REQLOC, and so on) for the interval beginning at 8:00:00. Messages are written to the job summary report.

OMEGAMON XE for DB2 PE calculates the next interval at 9:00:00. Accounting records at 9:12:11, 9:15:00, 9:30:01, 9:32:00, 9:43:00, and 9:55:59 are accumulated, consolidated and stored for the interval beginning at 9:00:00. Messages are written to the job summary report.

No Accounting records are written during the period from 10:00:00 to 12:00:00, so no intervals are calculated and no messages are written to the job summary report.

Data is stored for the following intervals:
- Beginning at 8:00:00 - containing three Accounting records
- Beginning at 9:00:00 - containing six Accounting records
- Beginning at 12:00:00 - containing two Accounting records
- Beginning at 13:00:00 - containing six Accounting records
- Beginning at 15:00:00 - containing one Accounting record
- Beginning at 16:00:00 - containing eight Accounting records

The systems programmer requested a report with FROM and TO times of 9:00:00 and 12:00:00.

The data that is available for reporting is restricted by the FROM and TO times that are specified in both GLOBAL and REDUCE. If you specify FROM and TO dates and times in GLOBAL, OMEGAMON XE for DB2 PE discards all records outside those dates and times before reducing. If you specify FROM and TO dates and times in REDUCE, all records outside those dates and times are not available to subsequent REPORT subcommands. In this example, no GLOBAL command is in effect and no FROM and TO dates or times are specified in the REDUCE subcommand.
Accounting records that are stored in intervals beginning at or later than 09:00:00, but less than 12:00:00 are accumulated, consolidated, and reported in an Accounting report. In this instance, the report contains Accounting records from 09:12:11 to 09:55:59.

The report header contains the following times:

```
INTERVAL FROM 07/14/08 09:00:00
   TO 07/14/08 10:00:00
REQUESTED ALL DATES 09:00:00
   TO 12:00:00
```

INTERVAL FROM and TO times indicate the actual content of the report. In this instance, the INTERVAL FROM time is 9:00:00, which is the first interval beginning at or later than the REQUESTED FROM time. The INTERVAL TO time is 10:00:00, which is the end time of the last interval beginning at a time less than the REQUESTED TO time.

The REQUESTED FROM and TO times from the REPORT subcommand are printed in the report, even if they are broader than the available data. In this case, ALL DATES is reported in place of an actual date because no date was specified in the REPORT subcommand.

If the requested reporting interval is broader than the available data, the INTERVAL FROM and TO dates and times actually reflect the content of the report. If no FROM and TO dates and times are specified in the REPORT subcommand, the FROM and TO dates and times from GLOBAL are used. If no FROM and TO dates and times are specified in GLOBAL, NOT SPECIFIED is printed.

In "Example 1" on page 151, the INTERVAL FROM and TO times are not the same as the REQUESTED FROM and TO times. However, the data encompassed by the INTERVAL FROM and TO times is the same as the data encompassed by the REQUESTED FROM and TO times. This is not always the case, as in "Example 2."

**Example 2**

```
ACCOUNTING
   REDUCE
      INTERVAL (60)
      BOUNDARY (30)
   REPORT
      FROM (,09:00:00)
      TO (,12:00:00)
```

This command is identical to the command in the "Example 1" on page 151 except that a BOUNDARY of 30 is included. The results are different.

The Accounting data starts at 8:31:05. The INTERVAL begins on the half hour. OMEGAMON XE for DB2 PE determines that the record at 8:31:05 falls into an interval beginning at 8:30:00 and sets the beginning interval time to 8:30:00. The following intervals are calculated:

- Beginning at 8:30:00 - containing five Accounting records
- Beginning at 9:30:00 - containing four Accounting records
- Beginning at 11:30:00 - containing two Accounting records
- Beginning at 12:30:00 - containing five Accounting records
- Beginning at 13:30:00 - containing one Accounting record
Beginning at 15:30:00 - containing two Accounting records
Beginning at 16:30:00 - containing seven Accounting records

An Accounting report is produced. The REQUESTED FROM and TO times are 9:00:00 and 12:00:00. Accounting records that are stored in intervals beginning at or later than 9:00:00, but less than 12:00:00, are included in the report. The intervals beginning at 9:30:00 and 11:30:00 are included in the report. No interval is calculated at 10:30:00 because there are no Accounting records.

In this instance, the report contains Accounting records from 9:30:01 to 12:17:54. Even though the REQUESTED FROM and TO times are the same as in [“Example 1” on page 151], the content of the report is different.

The report header contains the following times:

```
INTERVAL FROM 07/14/08 09:30:00
    TO 07/14/08 12:30:00
REQUESTED ALL DATES 09:00:00
    TO 12:00:00
```

In both [“Example 1” on page 151] and [“Example 2” on page 152], the INTERVAL FROM and TO dates and times are different from the REQUESTED FROM and TO times. However, this report contains data outside the REQUESTED FROM and TO times.

The INTERVAL FROM time is the start time of the first interval beginning at or after the REQUESTED FROM time (9:00:00 in this case), so the first interval that is reported starts at 9:30:00. Consequently, those records that are written between 9:00:00 and 9:30:00 are not included in the report.

The INTERVAL TO time is the end time of the last interval beginning before the REQUESTED TO time (12:00:00 in this case), so the last interval that is reported starts at 11:30:00. Consequently, those records that are written between 12:00:00 and 12:30:00 are included in the report. If the interval starts after the REQUESTED FROM time and before the REQUESTED TO time, all data from the interval is included, even if the interval extends beyond the REQUESTED TO time.

**Example 3**

```
Accounting REDUCE
    FROM (0,09:30)
    TO (0,13:00)
    INTERVAL (60)
    BOUNDARY (60)
    REPORT
    FROM (0,09:30)
    TO (0,13:00)
    SAVE

In this example, no GLOBAL command is in effect, so the FROM and TO dates and times that are specified in REDUCE are used.

REDUCE specifies that data is reduced at 60-minute intervals. BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the hour of the FROM time (09:00).
The first interval starts before the FROM time, but only contains data between the FROM time (09:30) and the start time of the next interval (10:00).

The following intervals are calculated:
- Beginning at 09:00:00 - containing four Accounting records
- Beginning at 12:00:00 - containing two Accounting records

**Note:** No intervals are calculated at 10:00:00 or 11:00:00 because there are no Accounting records.

The interval begin and end times are reported in the job summary report.

The systems programmer requested a report with FROM and TO times of 9:30:00 and 13:00:00. Accounting records that are stored in intervals beginning at or later than 9:30:00, but less than 13:00:00 are accumulated, consolidated, and reported in an Accounting report. In this example, the report only contains Accounting records from the interval starting at 12:00:00.

All reduced records are written to the Save data set (including the records from the interval starting at 9:00:00 that were excluded from the report).

The report header contains the following times:

```
INTERVAL FROM 07/14/08 12:00:00
   TO 07/14/08 13:00:00
REQUESTED ALL DATES 09:30:00
   TO 13:00:00
```

The INTERVAL FROM and TO times reflect the actual content of the report. The INTERVAL FROM time is the start time of the first interval beginning at or after the REQUESTED FROM time (9:30:00 in this case), so the first interval that is reported starts at 12:00:00. The INTERVAL TO time is the end time of the last interval beginning before the REQUESTED TO time (13:00:00 in this case).

The REQUESTED FROM and TO times from the REPORT subcommand are printed in the report, even if they are broader than the available data. In this case, ALL DATES is reported in place of an actual date because no date was specified in the REPORT subcommand.

In this example, the INTERVAL FROM and TO times are not the same as the REQUESTED FROM and TO times, although the data encompassed by both sets of times is the same. However, the content of the Save data set is different from the content of the report.

**The effect of REDUCE on TRACE:**

These examples show the effect of TRACE FROM and TO on REDUCE BOUNDARY and INTERVAL

**Example 1**

```
ACCOUNTING
REDUCE
   BOUNDARY (30)
   INTERVAL (60)
REPORT
   FROM (, 9:00:00)
```
The data that is available for tracing, as with reporting, is influenced first by the GLOBAL command. If you specify FROM and TO dates and times in GLOBAL, all records outside those dates and times are discarded before reducing or tracing.

The data that is available for tracing is also influenced by the REDUCE subcommand. If you specify FROM and TO dates and times in REDUCE, all records outside those dates and times are unavailable to subsequent TRACE subcommands. In this example, no GLOBAL command is in effect and no dates and times are specified for REDUCE, so all records in the input data are available to TRACE. The data that is available for tracing is not restricted by INTERVAL or BOUNDARY.

The ACCOUNTING TRACE subcommand specifies data from 12:00:00 to 17:00:00. The Accounting trace header contains the following times:

```
ACTUAL FROM 07/14/08 12:15:00
REQUESTED ALL DATES 12:00:00
                  TO 17:00:00
```

No dates are specified, so all dates are included. The REQUESTED FROM and TO times from the TRACE subcommand are printed in the trace.

**Example 2**

```
ACCOUNTING
   REDUCE
     FROM (10:00)
     TO (15:00)
   REPORT
     FROM (10:00)
     TO (15:00)
   TRACE
     FROM (09:00)
     TO (17:00)
```

In this example, the TRACE FROM time is before the REDUCE FROM time and the TRACE TO time is after the REDUCE TO time. The trace contains records that are written at or after 10:00:00 and before 15:00:00. The Accounting trace header contains the following times:

```
ACTUAL FROM 07/14/08 12:15:00
REQUESTED ALL DATES 09:00:00
                  TO 17:00:00
```

No dates are specified, so all dates are included. The REQUESTED FROM and TO times from the TRACE subcommand are printed in the trace. However, the available data is limited by the REDUCE FROM and TO times (10:00:00 to 15:00:00). The requested times do not reflect the actual content of the trace.

The ACTUAL FROM time is the timestamp of the first record in the trace. In this example, the first Accounting record after 10:00:00 is at 12:15:00. The trace includes the eight Accounting records between the REDUCE FROM and TO times.
The effect of REDUCE on FILE:

These examples show the effect of FILE FROM and TO on REDUCE BOUNDARY and INTERVAL.

Example 1

ACCOUNTING
REDUCE
  BOUNDARY (30)
  INTERVAL (60)
REPORT
  FROM (, 9:00:00)
  TO (,12:00:00)
FILE
  FROM (,12:00:00)
  TO (,17:00:00)

The data that is available for filing, as with reducing, is influenced first by the GLOBAL command. If you specify FROM and TO dates and times in GLOBAL, all records outside those dates and times are discarded before reducing or filing.

The data that is available for filing is also influenced by the REDUCE subcommand. If you specify FROM and TO dates and times in REDUCE, all records outside those dates and times are unavailable to subsequent FILE subcommands. In this example, no GLOBAL command is in effect and no times are specified for REDUCE, so all records in the input data are available to FILE. The data that is available for filing is not restricted by INTERVAL or BOUNDARY.

The ACCOUNTING FILE subcommand specifies data from 12:00:00 to 15:00:00. No dates are specified, so all dates are included.

Example 2

ACCOUNTING
REDUCE
  FROM (,10:00)
  TO (15:00)
REPORT
  FROM (,10:00)
  TO (15:00)
FILE
  FROM (,09:00)
  TO (17:00)

In Example 2, the FILE FROM time is before the REDUCE FROM time and the FILE TO time is after the REDUCE TO time. The file contains records that are written at or after 10:00:00 and before 15:00:00.

No dates are specified, so all dates are included. The data that is available for filing is limited by the REDUCE FROM and TO times (10:00:00 to 15:00:00). The requested times do not reflect the actual content of the file.

In Example 2, the first Accounting record after 10:00:00 is at 12:15:00. The file includes the eight Accounting records between the REDUCE FROM and TO times.
**Member-scope and group-scope reporting**

DB2 enables a query to be processed by several members of a data sharing group. Each member can split the work into parallel tasks. Accounting aggregates the parallel activity done in each member and includes it in the originating query activity.

The information in this section only applies to DB2 data sharing environments.

**Member-scope reports:**

Member-scope reports present the activity of a data sharing group by member.

The activity shown for each member consists of the originating query activity and any parallel activity, including any parallel activity that is performed on other members.

The data in member-scope reports is presented by a combination of location, group, subsystem, and member. Whenever one of these values changes, a new page is started and the page number is initialized. The following command produces a member-scope Accounting report shown in Figure 100. The command uses the SHORT layout and the default order of the OMEGAMON XE for DB2 PE identifiers, namely plan name within primary authorization ID.

```
; ACCOUNTING
; REPORT
; ...
```

![Figure 100. Member-scope Accounting report](image-url)
The data is ordered according to the authorization IDs and plans. All primary authorization IDs and plans from the input data are shown.

The TOTAL is printed for primary authorization IDs that contain more than one plan.

GRAND TOTAL is printed at the end of each member if there is more than one first-level identifier reported.

**Group-scope reports:**

Group-scope reports show the instrumentation data aggregated by the OMEGAMON XE for DB2 PE identifiers that you specified and by the individual members.
The data is presented by a combination of location and group. Whenever either of these values changes, a new page is started and the page number is initialized.

The following command produces a group-scope Accounting report shown in Figure 105. The command uses the SHORT layout and the default order of the OMEGAMON XE for DB2 PE identifiers, namely plan names within primary authorization IDs.

```
ACCOUNTING
   REPORT
      SCOPE(GROUP)
```

![Group-scope Accounting report](image_url)

Location: DSN420Y
Group: DSN420Y
Location: OMEGAMON XE for DB2 Performance Expert (V4R2)
Page: 1-1
Requested from: NOT SPECIFIED
To: NOT SPECIFIED
Order: PRIMAUTH-PLANNAME
Scope: GROUP
DB2 Version: V7
Scope: GROUP

```
PRIMAUTH
  PLANNAME  OCCURS  ROLLBK  SELECTS  INSERTS  UPDATES  DELETES  CLASS1  EL.TIME  CLASS2  EL.TIME
  MEMBER  DISRS  COMMIT  FETCHES  OPENS  CLOSES  PREPARE  CPUPROCESS  BUF.UPD  TOT.REF  LOCKOUT

USRT001
DSNTEP41 1 0 0.00 0.00 0.00 0.00 1:09:25:09.93 1:09:08:35.14 211.8K 114.00 47.00
Y42A 0 1 1.00 1.00 2.00 1:37.74:59.21 1:37.703681 4.30 6508.00 0

USRT001
DSNTEP42 10 5 0.00 0.00 0.00 0.00 20:33:32:48.22 14:30:612:146 823.10 658.00 0
Y42A 0 5 0.90 0.40 2.40 1:37:24:38.52 1:37:20:36.81 4.00 6508.00 0

USRT001
DSNTEP42 2 2 0.00 0.00 0.00 0.00 50.54:54:37 50.37:28:42 181.2K 10.00 13.00
Y42C 0 2 1.00 1.00 2.00 1:32:54:27:14 1:32:54:15:98.3 0.00 5676.00 0

USRT001
DSNTEP42 10 5 0.00 0.00 0.00 0.00 53.13:79:71 52.96:51:62 181.2K 10.00 13.00
Y42E 0 5 0.90 0.40 2.40 1:37:24:38.52 1:37:20:36.81 4.00 6508.00 0

USRT001
DSNTEP42 2 2 0.00 0.00 0.00 0.00 53.21:62:37 53.05:16:99 181.2K 10.00 13.00
Y42F 0 2 1.00 1.00 2.00 1:32:54:27:14 1:32:54:15:98.3 0.00 5676.00 0

*** GROUP TOTAL ***
USRT001
DSNTEP42 17 5 0.00 0.00 0.00 0.00 12:08:04:35:01 8:41:53:76:00 185.6K 37443.94 959.12
Y42A 0 12 533.6K 0.94 0.65 2.24 13:33:62:69:20 10:00:19:22:43 484.18 6208.35 0

*** TOTAL ***
USRT001
18 5 0.00 0.00 0.00 0.00 11:31:44:34:23 8:16:39:17:23 187.0K 35369.11 908.44
0 13 504.0K 0.94 0.67 2.22 12:53:86:21:58 9:32:27:62:12 457.50 6225.00 0

USRT002
DSNTEP41 1 1 0.00 0.00 0.00 0.00 36.29:56:13 36.18:59:47 51541.00 87.00 66.00
Y42A 0 0 0.00 1.00 0.00 0.00 56.79:52:1 56.78:52:73 11114.00 1790.00 0

USRT002
DSNTEP42 3 1 0.00 0.00 0.00 0.00 217.02:19:48 1:40:43:69:78 24010.67 6653.33 78.00
Y42A 0 2 200.2K 0.67 0.33 2.00 1:33:38:79:55 1:24:18:90:14 5253.67 627.00 0

*** TOTAL ***
USRT002
4 2 0.00 0.00 0.00 0.00 159.34:03:34 1:24:37:41:46 31493.25 5811.75 75.00
0 2 150.1K 0.75 0.25 2.00 1:54:24:00:37 1:17:33:60:29 6718.75 917.75 0

Figure 105. Group-scope Accounting report
The data is ordered according to the authorization IDs and plans. All primary authorization IDs and plans from the input data are shown. MEMBER is automatically added as the last ORDER identifier.

The GROUP TOTAL is printed for primary authorization IDs and plans that contain more than one member.

The TOTAL is printed for primary authorization IDs that contain more than one plan.

GRAND TOTAL is printed at the end of each group if there is more than one first-level identifier reported.

Statistics report set

Statistics reports and traces provide you with a way to analyze DB2 Statistics class 1 trace data.

DB2 accumulates statistics for the system services address space, database services address space, and DDF address space. These statistics are accumulated from the time DB2 is started until it is stopped. At a configurable interval (usually 30 minutes), DB2 logs the current statistics values using IFCID 1 and IFCID 2.

Use the Statistics report set to:
- View system-wide statistics for key DB2 components.
- Compare system performance over two or more reporting intervals.
- Assess system-wide performance for individual DB2 subsystems.
- Assess performance data for a group of data sharing DB2 subsystems.
- Summarize system performance data in a single report.

The Statistics report set provides the following functions:
- Traces present the difference (delta) between the Statistics recorded in two consecutive Statistics record pairs.
- Reports summarize Statistics data over one or more user-defined intervals.
- The File data set contains records in a format that are suitable for use by the DB2 LOAD utility.
- The Save-File utility changes Save data sets into sequential data sets for use by the DB2 LOAD utility.
- The REDUCE subcommand specifies intervals into which Statistics data is accumulated and apportioned. A report can then be produced that is sorted by these intervals.
The SAVE and RESTORE subcommands are used to consolidate Statistics for a number of Statistics record pairs and then save these record pairs in a Save data set for later restoration and further processing.

You can control the level of detail of a report by choosing one of the sample layouts or a layout that you have previously tailored.

You can use exception processing to identify entries with fields that contain values outside thresholds that you have previously specified.

In data sharing environments you can produce member-specific or group-specific reports.

The following topics provide additional information:

- General Statistics information

**General Statistics information**

This section contains information that is common to all Statistics functions. It also includes examples of Statistics processing.

**Statistics terms**

This section describes the various terms used in the Statistics report set.

**Delta and interval records:**

This section defines the differences among various DB2 Statistics counters, such as accumulated values, snapshot values, and high water mark values.

When a DB2 Statistics trace is active, DB2 maintains various Statistics counters and externalizes these counters at regular intervals (usually every 30 minutes). The main body of Statistics data is shown in IFCIDs 1 and 2. Although there is a small difference in their timestamps, you can assume that these IFCID are externalized simultaneously. These two records are called a *DB2 Statistics records pair*.

Other statistics information is shown in IFCID 199 (buffer pool statistics at data set level) and IFCID 225 (DBM1 storage statistics).

The counters provided in a DB2 Statistics records pair represent the DB2 activity between the time the pair is externalized and the time the DB2 system was last started. A counter is provided in one of the following forms:

- As an *accumulated value* since the DB2 system was last started. For example, the total number of SELECT statements that were executed since the system was last started.
- As a *current or snapshot value*. For example, the number of open data sets at the time the DB2 Statistics records pair was externalized.
- As a *maximum or high water mark value* the counter has reached since the time the system was last started. For example, the maximum number of open data sets at any time since the system was last started.

The Statistics report set does not report individual DB2 Statistics records pairs (for that purpose use the long Record Trace report). Instead, it calculates deltas between two consecutive DB2 Statistics records pairs and externalizes the delta records in Statistics traces and File data sets. It also uniformly distributes the delta records over user-specified intervals and externalizes the interval records in Statistics reports and Save data sets.

**OMEGAMON XE for DB2 PE delta records:**
The *delta record* is an OMEGAMON XE for DB2 PE term for a set of counters that describes the activity of a DB2 system between two consecutive DB2 Statistics records pairs.

For example, if a DB2 Statistics records pair is externalized at time \( t_1 \) and the next DB2 Statistics records pair is externalized at \( t_2 \), OMEGAMON XE for DB2 PE creates only one delta record, which represents the DB2 system activity between \( t_1 \) and \( t_2 \).

A counter in the delta record, like the counters in the DB2 Statistics records pairs, is provided in one of the following forms:

- **Accumulated value.** For example, the total number of SELECT statements that are executed between two consecutive DB2 Statistics records pairs. The accumulated value from a delta record is generally smaller than the accumulated value from the DB2 Statistics records pair which marks the end of the delta record and provides the values accumulated since the DB2 system was last started.

- **Current or snapshot value.** For example, the number of open data sets at the end of the delta record. It is the same as the value provided in the DB2 Statistics records pair which marks the end of the delta record.

- **The maximum or high-water mark value the counter reached from the time the system was last started until the end of the delta record,** for example, the maximum number of open data sets. It is the same as the value provided in the DB2 Statistics records pair which marks the end of the delta record.

When a delta record is calculated, OMEGAMON for DB2 PE externalizes it to Statistics traces and File data sets depending on your specification.

**Interval records:**

Interval record is a term for a set of counters that describes the activity of a DB2 system in a user-specified period of time.

Unlike the delta record, where the time interval is determined by two consecutive DB2 Statistics records pairs, you can specify the duration of an interval record by using the INTERVAL subcommand option of the REDUCE subcommand.

Interval records do not generally coincide with the delta records. Several delta records can be contained in one interval record, and vice versa. The interval records are not generally aligned with the delta records. The delta records are aligned with DB2 Statistics records pairs, while you can align the interval record by using the BOUNDARY subcommand option of the REDUCE subcommand.

When an interval record is calculated, OMEGAMON XE for DB2 PE externalizes it to Statistics reports and Save data sets depending on your specification.

To understand the relationship between the delta records and interval records, consider the following example.

The DB2 Statistics records pairs are generated every 30 minutes (their frequency is controlled by a DB2 system parameter). If a Statistics trace is started at 9:15 and stopped at 11:15, five DB2 Statistics records pairs are generated at the following times: 9:15, 9:45, 10:15, 10:45, and 11:15.
OMEGAMON XE for DB2 PE processes these records pairs and creates four 30-minute delta records starting at 9:15, 9:45, 10:15, and 10:45. You can print these delta records by specifying the TRACE subcommand or store these delta records in a data set that is suitable for loading into DB2 tables by specifying the FILE subcommand.

If you need a report by an hourly basis, you can specify an interval record duration of 60 minutes by using the INTERVAL subcommand option of REDUCE. By default, the interval records are aligned with hour boundaries. If you require a different alignment, use the BOUNDARY subcommand option. Based on this specification, OMEGAMON XE for DB2 PE uniformly distributes the already calculated 30-minute delta records into the corresponding interval records. Three 60-minute interval records are created starting at 9:00, 10:00, and 11:00. The first interval record contains the first delta record and a half of the second delta record. The second interval record contains a half of the second delta record, the third delta record, and a half of the fourth delta record. The third interval record contains a half of the fourth delta record. You can print these interval records by specifying the REPORT subcommand or store these interval records into a data set for later use by specifying the SAVE subcommand.

A counter in the interval record, like the counters in the delta records, is provided in one of the following forms:

- Accumulated value. For example, the total number of SELECT statements that were executed during the period of time specified for the interval record. Generally, this value is an approximation because the interval records are not aligned with the delta records. When an interval record crosses delta record boundaries, and vice versa, the values of the delta record counters are apportioned and uniformly distributed into overlapping interval records.
- The current or snapshot value is an approximation of the counter value at the end of the interval record. It is derived from the delta records’ current values and is weighted according to the overlap between the delta records and the interval records.
- The maximum or high-water mark value the counter has reached between the time the system was last started and the end of the last delta record included in the interval record calculation.

To produce a report that shows the Statistics data for each interval record, you first specify the interval-record duration and alignment with the INTERVAL and BOUNDARY subcommand options of the REDUCE subcommand, then the ORDER(INTERVAL) subcommand option of the REPORT subcommand.

A special type of the Statistics report is one where all DB2 Statistics records pairs in the input data set are consolidated in one interval record. Such a report is produced if no INTERVAL subcommand option is specified and the default INTERVAL(0) is assumed. In this case, the BOUNDARY and ORDER(INTERVAL) subcommand options do not apply. The start of the interval record is aligned to the first DB2 Statistics records pair, and there is no ordering by intervals because only one interval record is created. For example, if a Statistics trace is started at 9:15 and ended at 11:15, one interval record for that period is created that contains all counters pertinent to that interval, such as the number of INSERT statements executed from 9:15 to 11:15.

**Input for Statistics reports**
This section summarizes the DB2 IFCIDs, trace types, and classes for Statistics reports.
The following table summarizes the IFCIDs and DB2 trace classes required to produce Statistics reports and traces.

<table>
<thead>
<tr>
<th>Statistics information</th>
<th>DB2 IFCIDs</th>
<th>DB2 trace type and class</th>
</tr>
</thead>
<tbody>
<tr>
<td>System statistics, DB2 statistics</td>
<td>1, 2</td>
<td>Statistics, class 1</td>
</tr>
<tr>
<td>Buffer pool data set statistics</td>
<td>199</td>
<td>Statistics, class 8</td>
</tr>
<tr>
<td>DBM1 storage statistics</td>
<td>225</td>
<td>Statistics, class 6</td>
</tr>
</tbody>
</table>

Use the following DB2 command to collect all Statistics data:

```
-START TRACE (STATISTICS) CLASS (1,6,8) DEST (GTF) LOCATION (*)
```

Because the basic unit of processing in the Statistics report set is the delta record, at least two DB2 Statistics record pairs (IFCID 1 and 2) must be present in the input data set before statistics can be presented.

**Functions of the Statistics report set**

The Statistics report set consists of the REPORT, TRACE, FILE, SAVE, RESTORE, REDUCE functions and the Save-File utility.

**REPORT**

REPORT shows interval records that contain DB2 Statistics data aggregated over user-specified periods of time (see "Interval records" on page 162) which generally do not coincide with periods in which DB2 Statistics records are externalized. For example, you can produce reports that show DB2 system activity per hour, per day, or per the entire period in which the DB2 Statistics data is collected.

A special kind of the Statistics reports are exception reports in which only the interval records are presented that contain selected counters that exceed user-defined threshold values (see "Exception processing" on page 171).

The Statistics reports are produced for each DB2 subsystem and DB2 location present in the input data set. In a data sharing environment you can request the Statistics reports on a per-member basis or per-group basis where the Statistics data is aggregated across all members in the data sharing group (see "Member-scope and group-scope reporting" on page 166).

You can also tailor the format of the reports by specifying which report blocks of data and which fields from a report block are included in a report or by defining your own labels and headings associated with the reported fields (see "Controlling the level of detail in reports and traces" on page 165).

**TRACE**

TRACE shows delta records that contain DB2 Statistics data within periods of time marked by two consecutive DB2 Statistics records pairs (see "OMEGAMON XE for DB2 PE delta records" on page 161).

A special kind of the Statistics traces are exception traces in which only the delta records are presented that contain selected counters that exceed user-defined thresholds (see "Exception processing" on page 171).
The Statistics traces are produced for each DB2 subsystem and DB2 location present in the input data set. In a data sharing environment the Statistics traces are presented for each member of a data sharing group. Like reports, you can tailor the layout of the traces (see "Controlling the level of detail in reports and traces").

**FILE**

FILE stores delta records (the same data structures presented by the TRACE function) into a sequential data set that is suitable for use by the DB2 LOAD utility.

When delta records are in DB2 tables, you can produce tailored reports by using a reporting facility such as Query Management Facility (QMF). The FILE function can also be considered as an alternative way of archiving the DB2 Statistics data in Save data sets.

Checking for exception conditions is also available in the FILE function, in which case only the delta records are presented that contain selected counters that exceed user-defined thresholds (see "Exception processing" on page 171).

**SAVE**

SAVE stores interval records (the same data structures presented by the REPORT function) into a VSAM data set to:

- Archive the Statistics data for producing long-term reports
- Use the Save-File utility to create a sequential data set that is suitable for use by the DB2 LOAD utility.

**RESTORE**

RESTORE reloads a previously saved data set to report it with or without new Statistics data. Saved data can be restored and resaved as often as required.

**REDUCE**

REDUCE specifies the duration of the interval records that are to be presented in Statistics reports or stored in a Save data set.

You can also control the volume of data to be reported and saved by using the FROM and TO and the INCLUDE and EXCLUDE subcommand options.

**Save-File utility**

With the Save-File utility you can:

- Migrate Statistics Save data sets from earlier releases.
- Convert Statistics Save data sets into sequential data sets that are suitable for use by the DB2 LOAD utility.

**Controlling the level of detail in reports and traces**

You can specify the level of detail and the layout of Statistics reports and traces by using the LAYOUT subcommand option of the REPORT or TRACE subcommand.

The following sample layouts are supplied:

- **SHORT**
- **LONG**

In Statistics reports and traces, layout SHORT provides general data and layout LONG provides comprehensive data. Layout SHORT contains selected blocks and fields from statistics categories. layout LONG contains most, but not all, of the available blocks and fields.

If the use of commands, subcommands, and subcommand options does not provide sufficient control over the contents of reports and traces, you can use the...
User-Tailored Reporting (UTR) feature to create and tailor your own report and trace layouts. With UTR you can add, remove, and change individual fields and entire report blocks to control the volume, contents, and layouts of your reports and traces. For more information, see “Tailoring report layouts” on page 198.

**Member-scope and group-scope reporting**

DB2 collects the Statistics data on a per-member basis. DB2 Statistics traces have a local scope. To obtain the statistics for all members of a data sharing group, a DB2 Statistics trace has to be started at each of the members. These traces generally have different start and stop times and can have different periods in which the DB2 Statistics records are externalized.

The information in this section only applies to DB2 data sharing environments.

**Member-scope reports and traces:**

OMEGAMON XE for DB2 PE can process all members of a data sharing group at the same time and produce reports and traces that show the Statistics data on a per-member basis.

Such member-scope reports and traces are like those produced in non-data sharing environments. Reports and traces are produced on a per-subsystem basis.

The following extract from a long Statistics report demonstrates the concept of member-scope reporting:

- The data sharing group DSNDG0G consists of two members, DG1G and DG2G. Only the first page each member is shown.
- The statistics are presented separately for each member (note the MEMBER field in the page heading).
- The non-data sharing counters (such as the CPU TIMES block) as well as the data sharing counters (such as the Group Buffer Pools Activity block) are presented. To keep the figure compact only selected Statistics data is shown.
- The default Statistics interval is assumed, which means that the data is presented for the entire period the statistics are available.
- There is no aggregated statistics data for the entire group.
### Figure 107. Partial member-scope long Statistics report for DSNDG0G

**Chapter 5. Advanced reporting concepts**

<table>
<thead>
<tr>
<th>SQL DML</th>
<th>Quantity</th>
<th>/Second</th>
<th>/Thread</th>
<th>/Commit</th>
<th>SQL DCL</th>
<th>Quantity</th>
<th>/Second</th>
<th>/Thread</th>
<th>/Commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>27639.00</td>
<td>12.14</td>
<td>1151.63</td>
<td>244.59</td>
<td>LOCK TABLE</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>INSERT</td>
<td>12300.00</td>
<td>5.40</td>
<td>512.50</td>
<td>108.85</td>
<td>GRANT</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>UPDATE</td>
<td>1455.00</td>
<td>0.64</td>
<td>60.63</td>
<td>12.88</td>
<td>REVOKE</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DELETE</td>
<td>1145.00</td>
<td>0.50</td>
<td>47.71</td>
<td>10.13</td>
<td>SET HOST VARIABLE</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>PREPARE</td>
<td>2.00</td>
<td>0.00</td>
<td>0.08</td>
<td>0.02</td>
<td>SET CURRENT SOLID</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DESCRIBE</td>
<td>2.00</td>
<td>0.00</td>
<td>0.08</td>
<td>0.02</td>
<td>SET CURRENT RULES</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DESCRIBE TABLE</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>SET CURRENT PATH</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>OPEN</td>
<td>13915.00</td>
<td>6.11</td>
<td>579.79</td>
<td>123.14</td>
<td>SET CURRENT PRECISION</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CLOSE</td>
<td>13915.00</td>
<td>6.11</td>
<td>579.79</td>
<td>123.14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FETCH</td>
<td>3035.1K</td>
<td>1333.67</td>
<td>126.5K</td>
<td>26.9K</td>
<td>CONNECT TYPE 1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CONNECT TYPE 2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TOTAL</td>
<td>3105.5K</td>
<td>1364.59</td>
<td>129.4K</td>
<td>27.5K</td>
<td>RELEASE</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SET CONNECTION</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>STORED PROCEDURES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALL STATEMENT EXECUTED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>STATEMENT TRIGGER ACTIVATED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>PROCEDURE ABENDED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>ROW TRIGGER ACTIVATED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CALL STATEMENT TIMED OUT</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>SQL ERROR OCCURRED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CALL STATEMENT REJECTED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>USER DEFINED FUNCTIONS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXECUTED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>ABENDED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TIMED OUT</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>REJECTED</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Figure 107. Partial member-scope long Statistics report for DSNDG0G**
Group-scope reports:

In group-scope reports, OMEGAMON XE for DB2 PE applies the same user-specified Statistics interval to all members of a group, presents these Statistics intervals for each member, and then adds up the counters across all members and presents the counters as Statistics on a per-group basis.

Most Statistics counters are pertinent to the member that maintains them, for example, the counters that describe the local buffer pool activity. There are, however, some Statistics counters that are also pertinent to the entire data sharing group, for example, the counters that describes the global locking activity and group-buffer-pools usage. To present these counters on a per-group basis, OMEGAMON XE for DB2 PE combines the Statistics data of the individual members and presents it for the entire group. There are no group-scope traces because DB2 Statistics traces are not synchronized across a data sharing group. They have generally different start and stop times and frequency of externalizing the Statistics data.

The following extract from a short Statistics report demonstrates the concept of group-scope reporting.
The statistics are presented separately for each member (note the MEMBER field in the HIGHLIGHTS block).

Only selected counters are shown (HIGHLIGHTS, DATA SHARING LOCKS, and GROUP BP0), to keep the figure compact.

After the members’ Statistics are presented, the aggregated Statistics data for the entire group is shown.

- The MEMBER field in the HIGHLIGHTS block displays the number of members for which the Statistics are aggregated.
- The group’s Statistics counters are calculated by adding up the members’ Statistics counters.
- As the default Statistics interval is assumed (the data is presented for the entire period the statistics are available), the statistics are provided from the earliest to the latest time for which the statistics data is available.
- The group’s interval elapsed time is the average elapsed time of the members’ interval elapsed times.

Figure 109. Group-scope short Statistics report for DSGROUP1
### HIGHLIGHTS

<table>
<thead>
<tr>
<th>Interval Start</th>
<th>01/05/08 17:20:34.42</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval Elapsed</td>
<td>37:55.755771</td>
</tr>
<tr>
<td>Incremental Binds</td>
<td>0.00</td>
</tr>
<tr>
<td>DBAT Queued</td>
<td>0.00</td>
</tr>
<tr>
<td>Interval End</td>
<td>01/05/08 17:58:30.18</td>
</tr>
<tr>
<td>Outage Elapsed</td>
<td>0.000000</td>
</tr>
<tr>
<td>Auth Succ. w/out Catalog</td>
<td>10554.00</td>
</tr>
<tr>
<td>DB2 Command</td>
<td>38.00</td>
</tr>
<tr>
<td>Sampling Start</td>
<td>04/03/08 18:35:00.88</td>
</tr>
<tr>
<td>Total Threads</td>
<td>37.00</td>
</tr>
<tr>
<td>Buff. Updt/Pages Written</td>
<td>5.39</td>
</tr>
<tr>
<td>Total API</td>
<td>0.00</td>
</tr>
<tr>
<td>Sampling End</td>
<td>04/03/08 18:45:01.12</td>
</tr>
<tr>
<td>Total Commits</td>
<td>27038.00</td>
</tr>
<tr>
<td>PAGES Written/Write I/O</td>
<td>2.64</td>
</tr>
</tbody>
</table>

#### DATA SHARING LOCKS QUANTITY

<table>
<thead>
<tr>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLB CONT. RATE (%)</td>
<td>0.14</td>
</tr>
<tr>
<td>FLS CONT. RATE (%)</td>
<td>0.02</td>
</tr>
<tr>
<td>L-LOCKS XES (%)</td>
<td>N/C</td>
</tr>
<tr>
<td>LOCK REQ. (P-LOCK)</td>
<td>40764.00</td>
</tr>
<tr>
<td>UNLOCK REQ. (P-LCK)</td>
<td>40746.00</td>
</tr>
<tr>
<td>CHANGE REQ. (P-LCK)</td>
<td>0.00</td>
</tr>
<tr>
<td>SYNC. XES = LOCK</td>
<td>179.2K</td>
</tr>
<tr>
<td>SYNC. XES = CHANGE</td>
<td>101.4K</td>
</tr>
<tr>
<td>SYNC. XES = UNLOCK</td>
<td>175.3K</td>
</tr>
<tr>
<td>ASYN. XES-RESOURCES</td>
<td>0.00</td>
</tr>
<tr>
<td>TOTAL SUSPENDS</td>
<td>645.00</td>
</tr>
<tr>
<td>P-LCK/NFY ENG. UNAV</td>
<td>0.00</td>
</tr>
<tr>
<td>INCOM. RETAINED LCK</td>
<td>0.00</td>
</tr>
<tr>
<td>PSET/PART NEGOTIAT</td>
<td>0.00</td>
</tr>
<tr>
<td>PAGE NEGOTIATION</td>
<td>390.00</td>
</tr>
</tbody>
</table>

#### GROUP BP0 QUANTITY

<table>
<thead>
<tr>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP BP HIT RATIO (%)</td>
<td>13.15</td>
</tr>
<tr>
<td>SYN. READ(XI)-DATA RETURNED</td>
<td>31913.00</td>
</tr>
<tr>
<td>SYN. READ(XI)-NO DATA RETURN</td>
<td>2.00</td>
</tr>
<tr>
<td>SYN. READ(NF)-DATA RETURNED</td>
<td>45765.00</td>
</tr>
<tr>
<td>SYN. READ(NF)-NO DATA RETURN</td>
<td>158.0K</td>
</tr>
<tr>
<td>CLEAN PAGES SYN. WRTN</td>
<td>0.00</td>
</tr>
<tr>
<td>CHANGED PGS SYN. WRTN</td>
<td>172.2K</td>
</tr>
<tr>
<td>CLEAN PAGES ASYN. WRT</td>
<td>0.00</td>
</tr>
<tr>
<td>CHANGED PGS ASYN. WRT</td>
<td>1102.00</td>
</tr>
<tr>
<td>REG. PGS LIST (RPL) RQ</td>
<td>21518.00</td>
</tr>
<tr>
<td>CLEAN PGS READ RPL</td>
<td>9721.00</td>
</tr>
<tr>
<td>CHANGED PGS READ RPL</td>
<td>17652.00</td>
</tr>
<tr>
<td>PGS READ FROM DASD AFTER RPL</td>
<td>535.6K</td>
</tr>
<tr>
<td>ASYN. READ-DATA RETURNED</td>
<td>0.00</td>
</tr>
<tr>
<td>PAGES CASTOUT</td>
<td>73883.00</td>
</tr>
<tr>
<td>EXPLICIT X-INVALIDATIONS</td>
<td>0.00</td>
</tr>
<tr>
<td>CASTOUT CLASS THRESH</td>
<td>65.00</td>
</tr>
<tr>
<td>GROUP BP CAST. THRESH</td>
<td>1.00</td>
</tr>
<tr>
<td>CASTOUT ENG. UNAVAIL.</td>
<td>0.00</td>
</tr>
<tr>
<td>WRITE ENG. UNAVAIL.</td>
<td>0.00</td>
</tr>
<tr>
<td>READ FAILED-NO STOR.</td>
<td>0.00</td>
</tr>
<tr>
<td>WRITE FAILED-NO STOR.</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Figure 110. Group-scope short Statistics report for DSGROUP1 (continued)
Exception processing

Exception processing identifies Statistics report, trace, and File data set entries with fields that contain values outside the thresholds specified in the Exception Threshold data set. When exception processing is active, the data to be reported is checked against these thresholds.

There are two threshold types that you can set: warning and problem. A warning message is printed if a value is outside the first threshold and a problem message is printed if a value is outside the second threshold.

When exception processing is requested for a File data set, only delta records in exception status are included in the output data set.
Statistics exception reports, traces, and File data sets are identical to the usual Statistics reports, traces, and File data sets, except that they only contain interval records and delta records that have at least one field in exception status.

When delta records and interval records are processed during exception processing, values defined in the Exception Threshold data set are compared with the corresponding values in a delta or interval record. If any field is in exception status, the following occurs:

- If the JCL contains a valid DD statement for EXTRCDD1 or EXFILDD1, the identity of the delta or interval record (timestamp and subsystem identification) and the fields in exception status are logged in the exception logs.
- If you specified EXCEPTION with the REPORT, TRACE, or FILE subcommand, the interval or delta record is formatted and printed (or filed in case of the FILE subcommand) in the requested layout followed by the Exception Messages block of the report. An example is shown in Figure 112. The Exception Messages block of the report identifies the fields in exception status.

**Note:**

- The type of layout used has no effect on exception processing or on the content of the Exception Messages block of the report. All fields in exception status are reported. The layout used only affects the level of detail in the formatted interval or delta record. This means, the exception report can contain fields that have not been included in the corresponding trace or report.
- For group-scope reports, exceptions are checked only against the interval records that are aggregated across a data sharing group. However, member-specific statistics are shown wherever exceptions are found for the entire group.

The fields that caused the entry to be in exception status are identified along with the appropriate warning or problem message and are printed in the Exception Messages block of the report or trace.

Figure 112 shows an example of an Exception Messages block. It is printed after each interval or delta record that is found to be in exception status, provided that EXCEPTION was specified in the REPORT or TRACE subcommand.

```
<table>
<thead>
<tr>
<th>TYPE</th>
<th>FIELD ID</th>
<th>FIELD DESCRIPTION</th>
<th>BY VALUE</th>
<th>THRESHOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROBLEM</td>
<td>QBSTRI0</td>
<td>SYNCHRONOUS READS</td>
<td>TOTAL</td>
<td>53 &gt; 50</td>
</tr>
<tr>
<td>WARNING</td>
<td>QBSTRI0</td>
<td>SYNCHRONOUS READS</td>
<td>TOTAL</td>
<td>9 &gt; 1</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WARNING</td>
<td>QBSTRI0</td>
<td>SYNCHRONOUS READS</td>
<td>TOTAL</td>
<td>9 &gt; 1</td>
</tr>
<tr>
<td></td>
<td>BP7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Figure 112. Statistics Exception Messages report block**

The following columns are presented in the Exception Messages report block:

**TYPE** The type of exception. Valid values are WARNING and PROBLEM.

**FIELD ID**

The name of the field from the Exception Threshold data set.

**FIELD DESCRIPTION**

The description of the field.
**FIELD QUALIFIER**
The qualifier of the field. This is either the buffer pool ID, the group buffer pool ID, or the remote location. This field is blank if there is no field qualifier.

**BY**
The basis for the comparison. The following information can be printed in this column:

- **TOTAL**
  The threshold is specified as a “total” value.

- **MINUTE**
  The threshold is specified as a “by minute” value. The value in the interval or delta record is divided by the number of minutes in the interval or delta before making the comparison.

- **SECOND**
  The threshold is specified as a “by second” value. The value in the interval or delta record is divided by the number of seconds in the interval or delta before making the comparison.

- **COMMIT**
  The threshold is specified as a “by commit” value. The value in the interval or delta record is divided by the number of commits in the interval or delta before making the comparison.

- **THREAD**
  The threshold is specified as a “by thread” value. The value in the interval or delta record is divided by the number of threads in the interval or delta before making the comparison.

**VALUE**
The actual field value that is used for the comparison. For total comparisons, the value from the interval or delta record is printed. For “by minute”, “by second”, “by commit”, or “by thread” comparisons, the calculated “by minute”, “by second”, “by commit”, or “by thread” value is printed. The greater than symbol (>) or the less than symbol (<) is printed between this column and the THRESHOLD column, which indicates whether the value is larger or smaller than the threshold value.

**THRESHOLD**
The threshold defined in the Exception Threshold data set.

**Headers used in Statistics**
The headers in Statistics reports and traces are identical, except that INTERVAL FROM and INTERVAL TO in the report header is replaced by ACTUAL FROM in the trace header and the SCOPE field does not apply to Statistics traces.

An example of a Statistics report header is shown in Figure 113.

**Figure 113. Statistics report header example**

The Statistics report and trace headers contain the following information:

**LOCATION**
The DB2 reporting location. If the DB2 subsystem is installed without a
location name, either the DB2 subsystem ID (in non-data sharing environments) or the data sharing group name (in data sharing environments) is shown.

GROUP
The data sharing group that the DB2 subsystem belongs to.

MEMBER
The name of the DB2 data sharing member. This field is not printed for group-scope reports.

SUBSYSTEM
The ID of the DB2 subsystem that generated the data. This field is not printed for group-scope reports.

DB2 VERSION
The version number of the DB2 subsystem that generated the data.

Title - layout
The title of the report or trace and the layout. The layout can be the default layout (LONG or SHORT) or a layout that you have tailored yourself.

SCOPE
The scope of the report, which can be MEMBER or GROUP. A member-scope report shows a group’s instrumentation data member by member without merging the data. A group-scope report merges the instrumentation data that belong to the individual members and presents it for the entire group.

Note: All reports in a non-data sharing environment are member-scope reports.

EXCEPTION
Indicates an exception report or trace.

PAGE
The page number within the report in the format lll-nnnnnn, where lll denotes the sequence number of one or multiple locations (see LOCATION), and nnnnnn denotes the page number within the location.

REQUESTED FROM and TO
The FROM and TO dates and times that were specified in the REPORT or TRACE subcommand. If both FROM and TO dates and times are omitted from the subcommand, the FROM and TO dates and times from the GLOBAL command are printed.

If only the FROM date and time or only the TO date and time has been specified, NOT SPECIFIED is printed for the unspecified value. If FROM and TO are not specified in the subcommand and GLOBAL command, NOT SPECIFIED appears for both the FROM and TO values.

If you specified FROM and TO times without dates in the subcommand and GLOBAL command, ALL DATES is printed together with the specified times.

Note: Any FROM and TO times that are specified in REDUCE can affect the GLOBAL FROM and TO times.

ACTUAL FROM
The date and time of the first DB2 Statistics record that was processed by the trace.
INTERVAL FROM
The start date and time of the first interval record that is covered by the report.

INTERVAL TO
The end date and time of the last interval record that is covered by the report.

How values are reported
OMEGAMON for DB2 PE reports and traces show field values for both delta and interval reporting.

The field values are shown in the following forms:

TOTAL
The value is a total.

QUANTITY
The value is a total.

SECOND
The value in the interval or delta record is divided by the number of seconds in the interval or delta before it is reported.

COMMIT
The value in the interval or delta record is divided by the number of commits in the interval or delta before it is reported.

THREAD
The value in the interval or delta record is divided by the number of threads in the interval or delta before it is reported.

Exception processing
Exception reporting identifies Accounting and Statistics records that contain fields with values that are outside a desired range. This helps you to recognize performance problems in the DB2 subsystem and in threads.

You define exception thresholds for specific Accounting and Statistics fields in the Exception Threshold data set. If exception processing is active, the DB2 instrumentation data is checked against the threshold values in the Exception Threshold data set. Only records with at least one field value outside the thresholds are reported.

You can set the thresholds in the Exception Threshold data set by using exception profiling or with the help of the example Exception Threshold data set that is delivered with OMEGAMON XE for DB2 PE. Exception profiling can also be used to produce a report with details of the distribution and expected number of exceptions for each field.

Exception reports and traces are available in the Accounting report set and the Statistics report set. Each of these relates separately to Accounting or Statistics data.

In addition, the exception log lists both Accounting and Statistics exception records in the same report, in timestamp order. The contents of the Exception Log File data set are similar to the exception log. The Exception Log File data set can be used by the DB2 LOAD utility.
Exception processing output types

Exceptions are reported in various output types.

This topic describes the exception processing output:
- Accounting and Statistics exception reports
- Accounting and Statistics exception traces
- Accounting and Statistics Exception File data sets
- Exception log data set
- Exception Log File data set

Exception thresholds are specified in the Exception Threshold data set, ddname EXCPTDD. Therefore, the data set information for EXCPTDD must be specified in your JCL if you want to produce any of the listed output types.

Accounting and Statistics exception reports
An exception report is produced if you specify the EXCEPTION subcommand option in the REPORT subcommand.

Both report types contain entries that have at least one value outside the thresholds that you specified in the Exception Threshold data set.

After each report entry, information about the fields in exception status is printed.

Accounting and Statistics exception traces
You can produce an exception trace by specifying the EXCEPTION subcommand option in the TRACE subcommand.

Both trace types show records with fields values outside the thresholds that you specified in the Exception Threshold data set.

After each trace entry, information about the fields in exception status is printed.

Accounting and Statistics Exception File data sets
You can produce an example file data set by specifying EXCEPTION in the FILE subcommand.

The Exception File data set contains records that have at least one value outside the thresholds that you specified in the Exception Threshold data set.

Exception log data set
The exception log data set is produced automatically if you specified EXTRCDD1 in the JCL.

The exception log data set contains a list, in timestamp order, of DB2 Accounting and Statistics records with at least one field outside user-specified thresholds.

Exception Log File data set
The Exception Log File data set is written if you define the EXFILDD1 DD statement in the JCL.
The Exception Log File data set is a sequential data set that can be used with the DB2 LOAD utility.

The Exception Log File data set contains a list of Accounting and Statistics exception records similar to the list in the Exception Log data set.

To prevent the generation of the Exception Log File data set, omit the EXFILDD1 statement from your JCL (the preferred method), or specify DUMMY in the definition.

You can control the amount of data reported in the Exception Log File data set with GLOBAL INCLUDE or GLOBAL EXCLUDE, and FROM and TO.

Allocation values for EXFILDD1 are:

**RECFM:**
- VB

**LRECL:**
- 512

**BLKSIZE:**
- Recommended 4 096

### Exception Threshold data set

The Exception Threshold data set contains the exception thresholds for the Statistics and Accounting exception reports and traces. When exception processing is active, the instrumentation data is checked against these thresholds.

Threshold values in an Exception Threshold data set can be set or modified with the Exception Threshold data set editor (see "Specifying exceptions using the Exception Threshold data set editor" on page 59) or the exception profiling method (see "Exception profiling" on page 178).

A sample Exception Threshold data set is supplied in data set member RKO2DATA(DGOETV41). The sample contains a selection of exception fields with predefined threshold values and can be used to get started with exception reporting.

**Note:** Earlier versions of the sample Exception Threshold data set RKO2DATA(DGOETV41) contain entries with asterisks instead of predefined threshold values. Asterisks are intended to mark thresholds that are to be determined by the exception profiling method (described in "Exception profiling" on page 178). If you use these samples for exception reporting without performing exception profiling (which creates a new Exception Threshold data set with asterisks replaced by calculated values), the entries that contain asterisks generate warning messages during exception processing. In other words, earlier samples of the Exception Threshold data set are intended for exception profiling, later samples can be used for exception processing without modification.

If exception processing is started as part of the OMEGAMON Collector startup, where the Exception Threshold data set to be used is determined by the AUTOEXCPPTHNAME startup parameter, the Exception Threshold data set must be a sequential data set. For online monitoring, the Exception Threshold data set can be either a sequential data set or a member of a partitioned data set. If you create a new data set, preallocate it with the following attributes:
RECFM: 
  VB
LRECL: 
  ≥ 255
BLKSIZE: 
  6 233 or greater

Note: The sample Exception Threshold data set member RKO2DATA(DGOETV41) might have a different record length. When you copy member DGOETV41 to your newly allocated data set, you might get a warning that records are truncated. In this case, you can ignore this warning.

Related reading: For information about specifying and editing thresholds in an existing Exception Threshold data set, refer to Specifying exceptions using the Exception Threshold data set editor on page 59.

Related reading: For information about profiling an Exception Threshold data set with a sample of DB2 instrumentation data, refer to Exception profiling.

Exception profiling

You can use the exception profiling method to set adequate exception threshold values based on sample DB2 instrumentation data. This method eases the process of getting a suitable Exception Threshold data set that can be used for exception processing.

The basic procedure is to use an existing Exception Threshold data set, mark those fields that you want to be profiled by an asterisk (*), specify generalized warning and problem exception levels, and let the exception profiling batch job do the calculations based on the sample instrumentation data. The result is a new Exception Threshold data set with adequate exception threshold values.

Exception profiling can be repeated by using different sample data, or different field selections, or different generalized exception levels. An Exception Threshold data set can also be modified as described in Specifying exceptions using the Exception Threshold data set editor on page 59.

Data sets involved in exception profiling

Exception profiling requires several data set specifications.

The following data sets are essential for exception profiling. You need to specify these data sets in the exception profiling dialog:

- The input data set contains a sample of DB2 instrumentation data. The data is used to estimate your workload and decide which warning and problem thresholds to assign to the DB2 fields that are to be calculated.
  
  The more representative the data, the more accurate the calculation of the threshold values. The data should contain the type of data that you usually monitor. The input data should also contain a sufficient number of records to allow the profiling to be performed with reasonable confidence. The data should also cover an appropriate span of time.
  
  You can specify any combination of DPMOUT, SMF, or GTF data sets that contain DB2 instrumentation data.

- The input threshold data set is an Exception Threshold data set with some DB2 fields marked by asterisks (*). The asterisks serve as markers to indicate the DB2 fields for which exception profiling should calculate threshold values.
Exception profiling scans the *input threshold data set* for names of DB2 fields that have asterisks (*) assigned instead of threshold values. For these fields new threshold values are calculated based on sample data from the *input data set*. Fields that already have threshold values assigned are not considered and remain unchanged.

In other words, if you want exception profiling to calculate or recalculate certain threshold values, enter asterisks as threshold values for the corresponding fields in the *input threshold data set*.

- The *output threshold data set* is the new Exception Threshold data set. It is basically a copy of the *input threshold data set* (previously existing threshold values are retained), but previously marked fields obtained new threshold values.

The *output report data set* is optional. If you request a profile report in the exception profiling dialog, this data set will contain a report with details of the distribution and expected number of exceptions for each DB2 field listed in the new Exception Threshold data set.

**Exception profiling dialog**

This section describes the invocation and the dialog of exception profiling.

To use exception profiling, access the Interactive Report Facility (IRF) from the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS main menu and select option 6, *(Exception profiling)*. The Exception Profiling panel is displayed, as shown in [Figure 114](#).

Use the Exception Profiling panel to specify the profiling criteria required to calculate the threshold values and the required and optional data sets.

### DGOFEP00 Exception Profiling

Complete the following control information, then press Enter.

- **Warning exceptions**........... _____ (% of input data)
- **Problem exceptions**........... _____ (% of input data)
- **Produce profile report**......... _ (1=yes 2=no)

---

**Input data set**

---

**Input threshold data set**

---

**Output threshold data set**

---

**Output report data set**

---

**Command ==**>

F1=Help F2=Split F3=Exit F6=Browse F9=Swap F10=Global

F11=Inclexcl F12=Cancel

[Figure 114. Exception Profiling panel](#)

In this panel:

- Specify the percentage of input data that you want to be flagged as warnings (for example 5.00%) and the percentage of input data that you want to be flagged as problems (for example 2.50%). The percentages can be as small as 0.01%.
Choose whether you want to produce a profile report. The profile report documents the results of exception profiling (the expected number of exceptions for various thresholds).

Specify the name of the input data set that contains DB2 instrumentation data from your DB2 subsystem. It can be a GTF, SMF, or DPMOUT data set.

If you want to use several input data sets, you can concatenate them by editing the generated job stream by using option 2, Edit the generated job stream, on the Job Processing Selections panel (see Figure 115).

Specify the name of the input threshold data set that contains entries marked by asterisks (*) for the fields that you want to be profiled.

Specify the name of the output threshold data set that will contain the results.

If you requested a profile report, specify the name of the output report data set.

After you have completed the specifications, press Enter to generate the exception profiling job stream. The Job Processing Selections panel is displayed.

```
DG00JOBM Job Processing Selections
Command ===> ___________________________________________________________

Update the job statements as required, then select one of the following.

4  1. Browse the generated job stream
2. Edit the generated job stream
3. Store the job stream for future use
4. Submit the job stream for background execution

Job statement information:
//USERPO1P JOB (D01,CHAT), 'USR USERP21', MSGCLASS=V,CLASS=D, __________________________
// REGION=0M, NOTIFY=USERP01 ___________________________________________________________

Command ===> ___________________________________________________________
F1=Help  F2=Split  F3=Exit  F9=Swap  F12=Cancel
```

Figure 115. Submitting the exception profiling job

To submit the job, select option 4, enter your appropriate job statement information, and press Enter. Alternatively, you can browse, edit, or store the job stream for subsequent processing.

The new Exception Threshold data set is created.

Check the profiling report to ensure that the exception thresholds and the number of exceptions are satisfactory. If necessary, you can modify the threshold values in the new Exception Threshold data set, as described in “Specifying exceptions using the Exception Threshold data set editor” on page 59.

Now you can generate exception reports using the new Exception Threshold data set created by the exception profiling function.

**Exception profiling method**

Each record in the input data set is processed as it would be for normal exception processing. However, instead of checking each field for exception, the value of the
field is recorded. When all records have been processed, you can use the statistics in the profile report to determine the thresholds.

The thresholds are set to values that would generate the percentage of warning and problem level exceptions that you specified on the Exception Profiling panel (Figure 114 on page 179).

For example, suppose the field QIESECT is specified in the Exception Threshold data set for location DSNAPC5, with the operator set to greater than (>), and the warning and problem thresholds set to asterisk (*). The profiling criteria are specified as 5% for warning thresholds and 2.5% for problem thresholds on the Exception Profiling panel. If the data set is processed and 320 records are found for field QIESECT with the location DSNAPC5, exception profiling sets the warning threshold to the 17th highest record. This would generate 16 warning exceptions on average, which is 5% of the records (as you requested). Similarly, the problem threshold is set to the 9th largest record, averaging eight problem exceptions, which is 2.5% of the records.

**Exception profiling report**

Use the Exception profiling report to examine the results of exception profiling.

An example report is shown in the following figure.
The INPUT FROM and INPUT TO fields show the date and time of the first and last record in the input data set.

The profiling report also shows a table for each of the fields. These tables show the calculated threshold values (SPECIFIED THRESHOLD) and the expected number of exceptions (EXCEPTIONS GENERATED) for various exception percentages. These percentages are in multiples of 0.1, 0.25, 0.5, 1, 1.5, and 2 times the problem exception percentage specified on the Exception Profiling panel. For example, if the problem percentage is 5%, the table columns shown are 0.5%, 1.25%, 2.5%, 5%, 7.5%, and 10%.

Any error messages that are generated during exception processing are shown on the profiling report.

**Reporting distributed data**

OMEGAMON XE for DB2 PE can report activities that are associated with distributed work, where one DB2 subsystem accesses data from other DB2 subsystems.
DB2 uses the Distributed Data Facility (DDF) to allow an application program that is connected to one DB2 system to access data at a remote DB2 system, or any other relational database management system (DBMS) that supports DRDA.

OMEGAMON XE for DB2 PE can produce reports for a single DB2 location, or for a number of different DB2 host locations, when data is present. You can logically concatenate data sets from several locations in your JCL to produce a multilocation report. The data is then reported separately for each subsystem and ordered by location in alphabetic order.

When work is distributed across locations, the report shows activity at the reported location only.

The following information is reported for every location:
- Nondistributed transactions, this means, the allied threads at the reporting location.
- Local activity for distributed transactions that originate at the reporting location, this means, the allied-distributed threads at the reporting location without the corresponding DBATs at other locations.
- Remote activity performed at the reporting location as part of distributed transactions originating at other locations, this means, the DBATs at the reporting location.

Depending on the type of report, further detail information is reported.
- The Statistics report set shows following information in the Global DDF and the DRDA Remote Locations blocks of the report:
  - Distributed Data Facility (DDF) address space CPU times
  - Statistics for each DB2 remote location for DB2 host subsystems (MVS™, OS/390, and z/OS)
  - Aggregate Statistics for all locations that use DRDA
  - Other, nonlocation-specific DDF information
- The Accounting report set shows information for specific threads that participate in distributed activity. Information is shown for both requester locations and server locations. The reports show, for example, elapsed times spent at the server site and the number of transactions, conversations, SQL statements, rows, messages, and bytes sent from the requester and received by the server. This information is reported in the Distributed Activity block of the report.
- The SQL Activity report set shows all SQL data within threads. SQL statements are reported at the location where they were executed, both at the requester location and the server location.
- The Explain report set can show information for packages that are bound at a remote location.
  If a list of plans to be explained contains a remotely bound package on a DB2 host system, EXPLAIN automatically connects to the server and explains the remote package. Alternatively, you can specify the server location to which EXPLAIN is to connect and the plans and packages that you want explained.

The following topics provide additional information:
- “Selecting threads” on page 184
- “Examples of distributed transactions” on page 184
Selecting threads

You can select which threads or thread types are reported by using the INCLUDE and EXCLUDE subcommand options.

For example, if you want to produce reports that show only DBATs, you can include only threads that have a thread type of DBAT by using INCLUDE THREADTYPE(DBAT).

If you want to report the activity performed at a server location on behalf of a specific location, you can use the INCLUDE and EXCLUDE subcommands with REQLOC to only include data for that location.

If you want to report only distributed data, use EXCLUDE THREADTYPE(ALLIED) to exclude data for allied threads.

Examples of distributed transactions

The following examples show the types of distributed transactions and threads that are included in reports and traces.

Distributed transaction between DB2 host systems

This example shows distributed transactions between three DB2 host systems.

The following figure shows that all three locations are DB2 host systems that run on MVS, OS/390, or z/OS.

![Figure 117. Distributed transaction between DB2 systems](image)

This example shows a thread that originates at location A. Data is requested from location B and location C. Because this transaction is distributed across DB2 systems, it is called an allied-distributed thread at location A. The threads initiated at the remote locations to handle the client requests are called database access threads (DBAT).

If trace data is available, OMEGAMON XE for DB2 PE reports on the activity that is carried out at each location.
Distributed transaction between DB2 host systems and DB2 systems on other platforms

This example shows distributed transactions between DB2 host systems on different platforms.

The following example shows a host DB2 system that runs on MVS, OS/390, or z/OS, and two remote database management systems, such as DB2, that run on different operating systems, such as UNIX®.

OMEGAMON XE for DB2 PE reports on the activity that is carried out at Location A only.

Distributed transaction involving a DBAT-distributed thread

This example shows distributed transactions that involve a DBAT-distributed thread.

The following example shows three DB2 host systems.
When trace data is available for all three locations, a report shows data for the allied-distributed thread at location A, the distributed DBAT at location B, and the DBAT at location C.

## Reporting data sharing information

Data sharing gives individual DB2 subsystems full access to databases that are on shared hard disk drives. The DB2 subsystems that share the data belong to a data sharing group and each subsystem is considered a member of the group.

In a data sharing environment, you can monitor the performance of entire data sharing groups and individual members of a group. You can do this by generating reports or traces that combine performance information of all members, called *group-scope reports*, or by generating ordinary reports of individual members.

Group-scope reports are available in the Accounting, Locking, Audit, and Statistics report sets.

All report sets provide information about the performance of individual group members. In the report sets where group-scope reports are available, reports for individual members are called *member-scope reports*.

As with all aspects of performance, data sharing is best monitored by using exception processing. You can select exception thresholds for data-sharing-specific fields, and you can specify that the threshold is only checked for a certain group or member.

The following topics provide additional information:

- [Monitoring individual members](#)
- [Monitoring entire groups](#) on page 188
- [Collecting input data for group-scope reports](#) on page 190

### Monitoring individual members

You can monitor various aspects of performance for individual members of a group using any of the OMEGAMON XE for DB2 PE report sets.

#### Example of a member-scope Locking report

Locking reports are helpful in monitoring the locking of page sets.

If you want to monitor deadlocks and timeouts on shared databases, and if you want this information grouped by individual members, generate a Lockout report for every member. Because group-scope reports are available in the Locking report set, this report is called a member-scope report.

Member-scope reports also provide group-scope information because holders and waiters of lockups are shown. Member scope and group scope only influence the summarization of the report.

To generate a member-scope Locking report, specify the following command stream:

```none
: LOCKING REPORT
   LEVEL (LOCKOUT)
```
Member-scope report is the default, so you do not need to specify the SCOPE subcommand option. To order the report by page set within a database, specify the ORDER subcommand option.

The following example shows a member-scope Locking report for group DSHGRPXXX, which has two members, FIRST and SECOND. The report is two pages long because a new page is started when the member that is being reported changes.

The first page of the report shows the locking activity of the threads that have executed in member FIRST.

The second page of the report shows the locking activity of the threads that have executed in member SECOND.

Figure 120. Member-scope Locking Lockout report, page 1

Figure 121. Member-scope Locking Lockout report, page 2
Monitoring entire groups

Use group-scope reports to obtain an overall view of the performance of an entire group.

Group-scope reports are available in the Accounting, Locking, Audit, and Statistics report sets:

- The group-scope Accounting reports merge instrumentation data produced by the individual group members and present it for the entire group.
- The group-scope Locking reports provide a full picture of the locking activity within the entire data sharing group.
- The group-scope Statistics reports summarize group buffer pool and locking information for shared resources for all members. They also show key information, such as total number of threads and commits for an entire group.
- The group-scope Audit reports provide a comprehensive view of the access to shared resources by the users of the members of the group. For example, if you want a summary of users that belong to various members of a group who accessed, or attempted to access, page sets on shared databases, you can generate a group-scope Audit DML access report.

Example of a group-scope Locking report

In group-scope reports, events are aggregated by user-defined identifiers within the group, regardless of which member of the group actually generated the events.

To generate a group-scope Locking report, specify the following command stream:

```
... LOCKING
    REPORT
    LEVEL (LOCKOUT)
    SCOPE (GROUP)
...
```

The default order of a group-scope report is DATABASE-PAGESET.

The following example shows a group-scope Locking report for the same group, DSHGRPXX, as in the previous member-scope example. This report summarizes the lockout activity for both members FIRST and SECOND. The information is summarized by database, then page set, and lastly by individual member.
A group total is printed for the entire group when the database that is being monitored changes. **GRAND TOTAL** shows the timeouts and deadlocks in all databases for the entire group.

**Group-scope Statistics**

The group-scope Statistics reports show three categories of information summarized by group level.

**Highlights**

This category presents values such as the total number of threads and commits for the entire group.

**Data sharing locks**

This category presents locking information for shared resources for all members.

**Buffer pool data**

This category presents statistics per buffer pool summarized for all members of a group.

All other statistics data is presented in member-scope reports for detailed analysis on member level.
Group-scope Accounting

Group-scope reports show the instrumentation data aggregated by the OMEGAMON XE for DB2 PE identifiers that you specified and by the individual members.

The data is presented by a combination of location and group. Whenever either of these values changes, a new page is started and the page number is initialized.

Collecting input data for group-scope reports

To produce group-scope reports, you need input data from all members of a group.

For regular monitoring, you most likely collect the performance data into SMF data sets. If all DB2 subsystems of the data sharing groups reside in the same OS/390 system, you can collect the data to one SMF data set. If the members of the group reside on different OS/390 systems, you need to concatenate the SMF data sets from all OS/390 systems before you can use the data as input for group-scope reports.

Similarly, if you have collected instrumentation data for the individual members in separate data sets (GTF data sets or data sets created by collected report data), you can concatenate these data sets to generate group-scope reports.

Streamlining OMEGAMON XE for DB2 PE processing

Streamlining is the process of asking only for information in reports and traces that you really need.

OMEGAMON XE for DB2 PE is a comprehensive reporting tool, but for daily monitoring of DB2 you need a very limited amount of information. If you request a detailed report by using all input data that was gathered for a long time, OMEGAMON XE for DB2 PE processing takes up a lot of system resources. The result is pages of information that you are probably not interested in.

To avoid unnecessary performance overhead and to save time:

- Filter the input data, preferably by using the GLOBAL command.
- Disable OMEGAMON XE for DB2 PE internal sort if appropriate.
- Consider carefully how detailed a report you need.
- Define groups for identifiers that you want reported as a single entry.
- Use lists to simplify your command stream.
- Specify a REDUCE INTERVAL only if you want to report by intervals or produce several reports with different time spans.
- Define exception thresholds only for fields that you are interested in.
- Specify DPMOUT or keep a Save data set only if you are sure that you want to report the data again.
- Limit the number of SQL statements you want explained.

The following topics provide additional information:

- "Filter data" on page 191
- "Suppress OMEGAMON XE for DB2 PE internal sort" on page 194
- "Choose the right level of detail" on page 195
- "Group data" on page 195
- "Use lists" on page 196
Filter data

You can limit the amount of data to be processed by filtering the input data. You can specify filters in the GLOBAL command or in the REDUCE, REPORT, TRACE, or FILE subcommands.

You should specify the filters in GLOBAL, because only the data that passes through the GLOBAL filters is processed further. Ensure that you do not exclude records that are needed in subsequent processing.

**FROM and TO**

The simplest filter is the start and end time of the data to be reported.

Specify the start and end times by using the FROM and TO subcommand options.

*Example of reporting peak hour performance of a day:*

This example shows the use of FROM and TO within GLOBAL to report a specified date and time frame.

If you want to monitor the performance of your system only during peak hours and you want to produce the default versions of both, an Accounting report and a Statistics trace, you can specify:

```plaintext
GLOBAL
  FROM (05/15/08 ,08:30)
  TO (05/15/08 ,17:00)
ACCOUNTING
  REPORT
  STATISTICS
  TRACE

Both, the Accounting report and Statistics trace show information from 8:30 a.m. to 5 p.m. for the specified day.

*Example of reporting peak hour performance during a week:*

This example shows the use of FROM and TO within GLOBAL to report a specified time frame every day.

If the input data set contains data of at least a week, you can generate reports that show the performance of your system during peak hours for the whole week by specifying:

```plaintext
GLOBAL
  FROM (08:30)
  TO (17:00)
ACCOUNTING
  REPORT
  STATISTICS
```
Both, the Accounting report and Statistics trace show information from 8:30 a.m. to 5 p.m. for every day of the week.

**Example of generating an additional locking report:**

This example shows the use of FROM and TO for two purposes in the same command stream.

If you want to generate a Locking report for a day in the same job step, specify:

```
GLOBAL
  FROM (08:30)
  TO (17:00)
ACCOUNTING
REPORT
STATISTICS
TRACE
LOCKING
REPORT
  FROM (05/17/08)
  TO (05/17/08)
```

The Locking report contains data only from 8:30 a.m. to 5 p.m. for that day because no other data has passed the GLOBAL filtering.

**INCLUDE and EXCLUDE**

Another way to filter data is to include data only for particular OMEGAMON XE for DB2 PE identifier values, for example, user IDs or plans.

You can do this by using the INCLUDE and EXCLUDE subcommand options.

**OMEGAMON XE for DB2 PE identifiers:**

The identifiers describe the objects OMEGAMON XE for DB2 PE reports on.

The most commonly used OMEGAMON XE for DB2 PE identifiers describe:

- **The location**
  The LOCATION identifier is the name of the DB2 subsystem. If the input data contains data from several locations, you can include data only for those locations that you are interested in.

- **The user**
  OMEGAMON XE for DB2 PE uses two identifiers for the user ID. The first is the value of the authorization ID at the time of connection to DB2 (ORIGAUTH). The second is the authorization ID set at signon or identify (PRIMAUTH). For more information, see "Comparing original authorization IDs with primary authorization IDs" on page 234.
  For SQL requests from a client, the user ID of the user at the workstation is a possible identifier as well.

- **The plan**
  Use the PLANNAME identifier to select specific plans. Examples of plan names are DSNUTIL for utility, DSNBIND for bind activity, and the application plan name for CICS and IMS.
The package
Use the PACKAGE or MAINPACK identifiers to select plans and packages in the Accounting report set. Use PACKAGE to define specific packages, regardless of the plan to which they belong. Use MAINPACK to define plans that contain a specific package. For more information, see "Defining the MAINPACK identifier" on page 215.

The connection to DB2
You can select data for specific connections to DB2.
The connection ID (CONNECT) identifies the address space that interfaces with DB2. It can be, for example, the CICS or IMS ID.
The connection type (CONNTYPE) identifies the type of connection for a thread. It can be, for example, CICS, IMS-BMP, IMS-MPP, IMS-CNTL, or IMS-TBMP.

The correlation data
The correlation identifier identifies the DB2 task together with the connection ID. It is composed of two parts: the correlation name (CORRNAME) and the correlation number (CORRNMBR). This identifier can be very useful. For example, for CICS threads the correlation identifier contains the transaction ID. For more information, see "Correlation ID translation" on page 213.

Distributed activity
If you report data for distributed processing, you can select the locations that request the work (REQLOC) and the type of the threads (THREADTYPE) to be included, for example allied threads or DBATs.
In addition, the following identifiers of a client are supported to include or exclude related data:
– The end user’s user ID at the workstation (ENDUSER)
– The end user’s transaction name at the workstation (TRANSACT)
– The end user’s workstation name (WSNAME)

Example of reporting the performance of a specific location:

This example shows how only a single location identifier is included.
If the input data contains data from more than one location, but you only want to inspect the performance of location USIBMSNEWY11, specify:

```
GLOBAL
  INCLUDE (LOCATION(USIBMSNEWY11))
```

Only data for location USIBMSNEWY11 is processed.

Example of suppressing a specific connection type:

This example shows how a single location is included and a single connection type is excluded.
If you are not interested in the CICS activity for that location, specify:

```
GLOBAL
  INCLUDE (LOCATION(USIBMSNEWY11))
```
Only location USIBMSNEWY11 is reported and all connections except CICS are reported.

**Example of reporting about a specific user ID and plan:**

This example shows how a single authorization identifier and a single plan name is included.

If you suspect that authorization identifier USERID01 using plan NEWACC is causing a performance problem, you can report only data that belong to that user ID and plan. To do that, specify:

```
GLOBAL
  INCLUDE (AUTHID(USERID01)
  PLANNAME(NEWACC))
```

Only data with authorization ID USERID01 and plan NEWACC is reported.

**Suppress OMEGAMON XE for DB2 PE internal sort**

When you request Accounting functions only, it is often possible to avoid the OMEGAMON XE for DB2 PE internal sort of the input data. This reduces the size of the sort work files that must be allocated and reduces the processing time.

The PRESORTED option of the GLOBAL command controls the internal sort. It has the following values:

- **NO**  
  Do not disable the sort. This is the default.

- **ENFORCE**  
  Disable the sort, but terminate processing if out-of-sequence records are present. Use this option only if the input data set has been sorted, for example, when you reprocess data from the DPMOUT data set.

- **ACCEPT**  
  Disable the sort and accept out-of-sequence records. By using this option it is possible to create complete and accurate Accounting reports (including records from multiple locations) from data sets that have not been sorted, for example SMF or GTF. The following limitations apply with this option:
  - Some checking of the input data is not performed. Ensure that the concatenation of two or more data sets does not result in duplicate records, for example, if trace data was collected in two data sets during the same period.
  - Only one location is reported per trace. To report more locations, specify multiple TRACE subcommands with different INCLUDE subcommand options for each location.
  - Trace entries might not be printed in time sequence. If a trace contains entries that are out of sequence, a message that shows the number is displayed at the end.
  - If INTERVAL(0) is in effect (the default), the interval times shown in a report heading might not be accurate. Ordering reports by interval should be avoided when the interval is zero.
For example, to produce an Accounting report without sorting the input data, specify:

```
GLOBAL
  PRESORTED(ACCEPT)
ACCOUNTING
```

The resulting Accounting report shows Accounting data for all locations in the input data set, without performing an internal sort.

**Choose the right level of detail**

Before you produce a report or a trace, consider how much detail you need. Use the LAYOUT subcommand option of the Accounting and Statistics reports and traces to control the amount of data to be produced.

In most situations the default layouts, which are short versions of reports and traces, provide enough detail for monitoring and problem determination. Do not use the most comprehensive layouts that show all possible fields unless you need this information.

If the use of commands, subcommands, and subcommand options does not provide sufficient control over the contents of reports and traces, you can use the User-Tailored Reporting (UTR) feature to create and tailor your own report and trace layouts. With UTR you can add, remove, and change individual fields and entire report blocks to control the volume, contents, and layouts of your reports and traces. For more information, see "Tailoring report layouts" on page 198.

**Group data**

Use the GROUP command to define a set of OMEGAMON XE for DB2 PE identifier values that can be used when you request certain reports.

The information for the set is reported as a single entry in the reports.

In GROUP processing the data for all members of the set is consolidated into one record. This improves OMEGAMON XE for DB2 PE performance because fewer records need to be processed.

Sets are also useful for reporting purposes such as if you want to report data for an entire department instead of every individual person.

This is how you specify sets. Assume that your sales department consists of three users, USER001, USER002, and USER003. You want to produce an Accounting report that shows performance data for that department. You can enter:

```
GROUP
  (PRIMAUTH(SALES(USER001,
                USER002,
                USER003)))
ACCOUNTING
  REDUCE
    INCLUDE (PRIMAUTH(G(SALES)))
REPORT
```
ORDER (PRIMAUTH)

The Accounting report shows information for the sales department as a single entry.

You can use GROUP with all report sets except Audit, Record Trace, and Statistics. The most common identifiers used to group data are:
- ORIGAUTH and PRIMAUTH
- PLANNAME
- CONNECT and CONNTYPE
- CORRNAME and CORRNMBR

**Use lists**

Use the LIST command to define a list of values for an OMEGAMON XE for DB2 PE identifier that can be used in INCLUDE and EXCLUDE, instead of individually entering each member. The members of the list are treated as if they were entered individually.

LIST processing does not affect OMEGAMON XE for DB2 PE performance, but can make it easier to specify and read your command stream.

Assume that you want to see how some plans that are used by the sales department affect performance. To do that, produce an Accounting report and an SQL Activity trace. If you specify a list for all plan name values, as in the following example, you can use the list name (SALES) in all commands in that job step.

```
LIST (PLANNAME(SALES(PLAN001,
                   PLAN002,
                   PLAN003,
                   PLAN004,
                   PLAN005,
                   PLAN006,
                   PLAN007,
                   PLAN008)))

ACCOUNTING
REPORT
   INCLUDE (PLANNAME(L(SALES)))
   ORDER (PLANNAME)
SQLACTIVITY
TRACE
   INCLUDE (PLANNAME(L(SALES)))
```

Both, the Accounting report and the SQL Activity trace show information for all individual plans that are specified in the list.

You can use LIST with all report sets. You can specify lists for any OMEGAMON XE for DB2 PE identifier values that are allowed in that report set.

**Be careful with INTERVAL**

If you want to use OMEGAMON XE for DB2 PE for trend analysis, you probably want to report data by interval. To do this, first reduce the input data to your Accounting and Statistics reports by using an interval other than the default zero. Then order the report by intervals.
For example, if you want to report the data at daily intervals, specify INTERVAL (1440) in the REDUCE subcommand and ORDER(INTERVAL) in the REPORT subcommand.

Interval processing affects the performance of the job. If you do not intend to produce reports by interval, accept the default (0) for INTERVAL, which means that no interval processing is performed.

**Specify only relevant exception thresholds**

Exception processing is the most effective way to identify system performance problems. You can set exception thresholds for virtually all Accounting and Statistics fields. However, carefully consider the fields for which to specify exception thresholds. The more fields you specify, the greater the effects on processing.

Specify exception thresholds only for those fields that you believe will signal poor performance in your environment. For information about how to generate exception reports and for a list of threshold fields that you should use, see "Using Statistics and Accounting reports to identify exceptions" on page 58.

**Do you need a DPMOUT data set?**

You can keep the preprocessed input data in the output data set DPMOUT. The DPMOUT data set can be used as input to OMEGAMON XE for DB2 PE.

If you do not specify DPMOUTDD, only the records required for the current job step are processed, which improves system performance.

**Do you want to save reduced data?**

If you need to keep historical Accounting and Statistics data about DB2 performance, consider using REDUCE and SAVE.

This processing has advantages and disadvantages. Reducing and saving data uses a considerable amount of system resources, but the resulting Save data set is much smaller than the original input data set. The size of the resulting Save data set varies with the specified reduction interval and the type of environment the data is from (for example, the number of different users and plans present in the input data).

You can produce reports from the reduced and saved data by using the RESTORE command. Remember that you cannot produce traces from reduced data.

**Limit statements to be explained**

When you produce Explain reports, limit the number of SQL statements to be examined.

There are various ways you can limit the output. You can specify only the plans or packages that you are interested in, or you can use the LIMIT, PACKLIMIT, and PACKAGES NO options.

To reduce attachment costs, it is always best to group the subsystems you are reporting.
Customizing OMEGAMON XE for DB2 PE functions

Customizing is the process of tailoring OMEGAMON XE for DB2 PE functions for specific needs, for example, specifying thresholds or modifying report layouts.

There are some OMEGAMON XE for DB2 PE functions that you can customize to meet your particular needs. The things you can tailor are:

- Exception thresholds
  Exception reporting identifies DB2 threads and Statistics intervals with fields values outside defined limits. You can either specify the thresholds yourself (for more information see "Using Statistics and Accounting reports to identify exceptions" on page 58) or you can let OMEGAMON XE for DB2 PE fill these values (for more information see "Exception profiling" on page 178).

- Report layouts
  Several model layouts for Accounting and Statistics reports are provided with the product. If none of them meets your needs, you can tailor them or create your own report layouts.

- Time zone specifications
  You can modify the times used in reporting if the CPU clock of your OS/390 system is not set to the local time or if you want to report data from two or more systems that have different CPU clock settings.

- The correlation name and number
  OMEGAMON XE for DB2 PE provides a default translation of the DB2 correlation ID for most environments. You can change the way the DB2 correlation ID is translated into correlation name and correlation number.

- MAINPACK identifier
  MAINPACK is an OMEGAMON XE for DB2 PE identifier that you can use to identify a plan by the first or the last package within the plan. You can also define whether to use the package ID, the collection ID, or the location from the package name to identify the main package.

- Exception field descriptors
  You can modify exception field descriptors by using the ISPF editor.

Changes to these functions are recorded in the DPMPARMS data set. The data set needs to be allocated. You can create multiple DPMPARMS data sets and you can concatenate several DPMPARMS data sets as required.

The following topics provide additional information:

- "Tailoring report layouts"
- "Specifying time zones" on page 209
- "Correlation ID translation" on page 213
- "Defining the MAINPACK identifier" on page 215
- "Modifying exception field descriptors" on page 221

Tailoring report layouts

OMEGAMON XE for DB2 PE supplies several Accounting and Statistics model reports and traces. You can create your own layouts and you can tailor layouts with the User-Tailored Reporting feature (UTR).

With UTR you can:
• Add entire blocks and individual fields to an existing layout, for example, to include some additional fields or entire blocks of related fields in a report or trace.

• Remove entire blocks and individual fields from an existing layout, for example, to exclude some fields or entire blocks of related fields that are not of interest to you from a report or trace.

• Change the relative positions of blocks and fields in an existing layout, for example, to rearrange blocks and fields that are of particular interest.

• Change block and field labels, for example, to abbreviate labels to keep reports compact, or to spell out block and field names, or to use acronyms.

Example of tailoring an Accounting report
This example describes how to add a block of fields to an existing report layout and how to replace some of the existing fields with new fields.

Suppose that the standard layout of the short Accounting report (supplied as Accounting report layout SHORT) does not provide the information that you need to monitor your DB2 installation and its specific workload. The standard layout looks like this:

Assume that you want to include more buffer pool information in your layout, as in the following tailored report layout.

Figure 123. Accounting report - standard layout

Assume that you want to include more buffer pool information in your layout, as in the following tailored report layout.
The tailored layout has been named BUFFER and contains buffer pool information in addition to General Accounting data. The buffer pool fields in the General Accounting block have been replaced with other fields to avoid duplicate information. In this example, class 3 suspension times and events are shown.

The following steps describe how to produce the tailored layout.

Selecting a model layout:

These steps describe how to select an existing report layout as the basis for a new layout that is to contain tailored information.

About this task

1. From the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS main menu, select option 5 (Customize report and trace layouts).
   The User-Tailored Reporting Layout Generation panel is displayed.
2. For this example, select option 1 (Accounting report) from the menu, enter the name of the DPMPARMS data set where the tailored layout is to be stored, and press Enter.

   The UTR Layout Selection panel is displayed.

3. For this example, select layout SHORT as the basis for your layout by entering a selection character (/) in the input field next to the layout name. Always choose the model that is most similar to the layout you want.

   The UTR Block Selection panel for the Accounting Report SHORT layout is displayed.

Figure 125. User-Tailored Reporting - selecting an Accounting report as model layout

Figure 126. User-Tailored Reporting - selecting a short Accounting report as model layout

3. For this example, select layout SHORT as the basis for your layout by entering a selection character (/) in the input field next to the layout name. Always choose the model that is most similar to the layout you want.

   The UTR Block Selection panel for the Accounting Report SHORT layout is displayed.
The Order column shows which blocks are included in the selected layout. In this example, the blocks labeled GENERAL, PACKAGE GENERAL, and DISTRIBUTED ACTIVITY are included. The numbers in the Order column indicate the sequence in which the blocks are printed. Fields that have not been selected are numbered 99999.

Adding a block:

These steps describe how to add a report block to a selected report layout that is to contain tailored information.

About this task

In this example, the BUFFER POOL ACTIVITY block is added to the selected layout.

1. In the UTR Block Selection panel, use F8 to scroll to the label BUFFER POOL ACTIVITY. Notice that there are two blocks with this label. Press F5 to display a brief description of the block's content, as shown in the following figure. In this example, the column form is chosen.
2. Enter a number in the Order column next to the block label to determine the relative position of the added block to the existing blocks.

   In this example, a number between 10 and 20 determines that the BUFFER POOL ACTIVITY block is printed after the GENERAL block and before the PACKAGE block. A number less than 10 would add the BUFFER POOL ACTIVITY block before the GENERAL block.

3. To see which fields can be included in the added block, enter a selection character (/) in the Modify column next to the block label.

   The UTR Field Selection panel is displayed.
Similar to the UTR Block Selection panel, use F7 (Up) and F8 (Down) to scroll through the list of fields and F5 (Extend) to display extended field descriptions. The Order column determines the order of the fields within a block. A value of 99999 means that a field is not selected.

4. For this example, assume that all fields of the BUFFER POOL ACTIVITY block are selected by default. Because no fields need to be added or deleted, press F12 (Cancel) to return to the UTR Block Selection panel.

**Replacing fields:**

In this example, the duplicate fields in the GENERAL Accounting report block are replaced with class 3 suspensions and class 3 suspension times.

**About this task**

You might have noticed that the Buffer Pool block contains some fields (GETPAGES, SYN.READ, BUF.UPDT, and TOT.PREF) that are also in the General Accounting block (see Figure 123 on page 199). To modify the fields in the General Accounting block, perform the following steps:

1. In the UTR Block Selection panel, type a forward slash (/) in the Modify column next to GENERAL and press Enter.
   
   The UTR Field Selection panel is displayed.
2. In the UTR Field Selection panel, use F8 (Down) to find the fields that you do not want in the General Accounting block. Overtype the numbers (170, 180, 190, and 200) in front of the fields with blanks and press Enter (to reorder the remaining fields).

3. Use F7 (Up) and F8 (Down) to find the fields that you want to add.

---

**Figure 130. User-Tailored Reporting - deleting fields**

2. In the UTR Field Selection panel, use F8 (Down) to find the fields that you do not want in the General Accounting block. Overtype the numbers (170, 180, 190, and 200) in front of the fields with blanks and press Enter (to reorder the remaining fields).

3. Use F7 (Up) and F8 (Down) to find the fields that you want to add.
4. For this example, enter any number from 160 to 210 in the Order column in front of the fields that you want to add and press Enter.

5. Scroll back to see that the newly inserted fields appear at the position that you intended.

You do not need to delete existing fields when you add new fields. You can insert new fields between existing fields or place new fields before or after existing fields. Also, you can delete fields without replacing them with new fields. When the layout is generated the gaps are automatically filled in.

There is no limitation to how many fields or blocks can be selected in a layout. However, carefully consider what information you really need to prevent your reports from getting too long.

**Displaying the layout:**

These steps describe how to display the modified layout.

**About this task**

1. Press F3 (Exit) to return to the UTR Block Selection panel.
2. Press F6 (Browse) to display the new layout.
   A sample report using the modified layout is displayed.
### Saving the layout:

These steps describe how to save the modified layout after you have completed the changes to your layout.

1. Use F3 (Exit) as required to return to the UTR Block Selection panel.
2. If you do not want to save the layout, press F12 (Cancel) to leave the UTR Block Selection panel.
3. To save the layout, press F3 (Exit).

The UTR Save Layout Details panel is displayed.

---

**Figure 132. User-Tailored Reporting - browsing a sample layout**

3. You might need to scroll right (F11) and down (F8) to see the entire report. 

   **Figure 132** shows the right-hand side of the sample report.

---

**ACCOUNTING REPORT - SHORT**

<table>
<thead>
<tr>
<th>TOP NUMBER REQUESTED:</th>
<th>LECTS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>CLASS1</th>
<th>EL. TIME</th>
<th>CLASS2</th>
<th>EL. TIME</th>
<th>CL3</th>
<th>SUSP</th>
<th>LOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>SQLSTM</td>
<td>CL7 CPU TIME</td>
<td>CL8 SUSP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#OCCURS</td>
<td>CL7 ELAP.TIME</td>
<td>CL8 SUSP.Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OMEGAMON XE for DB2 Performance Expert (V4R2) PAGE:**

Accounting report - short requested from:

ACCOUNTING REPORT - SHORT

Requested from:

TO:

Interval from:

TO:

Top number requested:

<table>
<thead>
<tr>
<th>LECTS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>CLASS1</th>
<th>EL. TIME</th>
<th>CLASS2</th>
<th>EL. TIME</th>
<th>CL3</th>
<th>SUSP</th>
<th>LOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>SQLSTM</td>
<td>CL7 CPU TIME</td>
<td>CL8 SUSP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#OCCURS</td>
<td>CL7 ELAP.TIME</td>
<td>CL8 SUSP.Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**OMEGAMON XE for DB2 Performance Expert (V4R2) PAGE:**

ACCOUNTING REPORT - SHORT

Requested from:

TO:

Interval from:

TO:

Top number requested:

<table>
<thead>
<tr>
<th>LECTS</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETES</th>
<th>CLASS1</th>
<th>EL. TIME</th>
<th>CLASS2</th>
<th>EL. TIME</th>
<th>CL3</th>
<th>SUSP</th>
<th>LOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>SQLSTM</td>
<td>CL7 CPU TIME</td>
<td>CL8 SUSP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#OCCURS</td>
<td>CL7 ELAP.TIME</td>
<td>CL8 SUSP.Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Type a new name for the new layout in the Layout field and a descriptive text in the Description field.
   To keep the existing layout SHORT, and create an additional layout, give the layout a new name by typing over the existing name. You can also type over the existing description.
   You can save the modified layout with the name of the layout you based it on. However, if you do this, the documentation does not reflect your new layout. If you pressed Enter, you would save the modified layout with the name SHORT. In this example, the layout is named BUFFER and the description is changed.

5. Specify whether you want each new record on the report or trace to start on a new page.
   If your layout contains several blocks of information, it is often clearest to have every entry start on a new page.
   If your layout only contains a few blocks of information, several entries most likely fit on the page. Also, even if your layout is long, but you intend to use it with options, such as ORDER(PACKAGE), that limit the amount of information produced, you probably want more than one entry per page.

6. Press Enter.
   The Accounting report layout BUFFER is saved into your DPMPARMS data set.

The layout change is now complete. The UTR Layout Selection panel is displayed. Notice that the layout that you just created is shown in the list of layouts. To exit from this panel, press F3.

**Using the layout:**

These steps describe how to use the modified layout.

**About this task**
1. Specify the name of your DPMPARMS data set in the JCL.
2. You can now run Accounting reports using the new layout.

```plaintext
ACCOUNTING
  REPORT
```
Specifying time zones

The TIMEZONE option of the GLOBAL command provides a means of adjusting the times of the data to be reported.

You need to adjust the times if:

- The CPU clock of your OS/390 system is not set to the local time, but you want to use the local time in your reports.
  The CPU clock can be set to Greenwich Mean Time (GMT) or to the local time of another location, for example the local time of your head office.
- You want to generate reports or traces that show activity at more than one location and the CPU clock settings of the locations are different. This is often the case when the locations are in different time zones.

The data for calculating the adjusted times is stored in the LOCDATA member of the DPMPARMS data set. You can enter and edit the data using the Interactive Report Facility (IRF).

Time Zone Data Editor

Time zone information is entered with the Time Zone Data Editor, which is accessed through the Interactive Report Facility (IRF).

About this task

To access the editor, select option 4 (Maintain parameter data sets) from the Performance Expert main menu. The Data Set Maintenance Menu is displayed.

![DGOPMENU Data Set Maintenance Menu](image)

Figure 134. Selecting Maintain Time Zone Information

Type 3 in the input field to select Maintain time zone information, and type the name of your DPMPARMS data set on the line below DPMPARMS data set.

Press Enter to go to the Time Zone Data Editor panel.
Use the editor to enter time zone specifications for the reporting locations. Specify the location, the geographical time zone, and the CPU clock setting of the z/OS system under which the DB2 subsystem runs.

<table>
<thead>
<tr>
<th>DGOPPLDS</th>
<th>Time Zone Data Editor</th>
<th>Row 1 to 1 of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direction (1=East 2=West)</td>
<td>Action</td>
<td>Location -- Time Zone -- -- CPU Clock --</td>
</tr>
<tr>
<td>Direction HH MM</td>
<td>Direction HH MM</td>
<td>-- -- -- --</td>
</tr>
</tbody>
</table>

Enter the required information in the input fields. You can modify existing information by typing over it.

**Direction**
Indicates the direction of the location relative to Greenwich:
1 For locations east of Greenwich
2 For locations west of Greenwich

**Action**
The standard ISPF editor line commands. For example:
I To generate a new line
D To delete a line
R To duplicate a line
M To move a line

**Location**
The name of the location. You would usually use the location identifier of the DB2 subsystem. Enter an asterisk (*) to specify the default time zone value. You can specify only one default value.

**Time Zone**
The difference in hours and minutes between the geographical local time and GMT.

**CPU Clock**
The difference in hours and minutes between the CPU's Store Clock instruction value of the location and GMT.

You can get information about each entry field by moving the cursor to the field and pressing F1 (Help) to display the help text for the field.

**Examples of time zone processing**
In the following examples, the settings in the Time Zone column show the real time zones relative to GMT. These settings can be maintained to reflect seasonal differences in daylight saving.

The values in the CPU clock column are for illustration purposes only. You need to find out the CPU clock settings of the locations you want reported because every site can decide whether to set the CPU clock to local time, GMT, or some other value.
Example 1: Time zones:

This example shows how a specific location is assigned to a time zone and how this location's time is reflected in reports.

The figure shows the contents of the LOCDATA member of DPMPARMS. In this example, we want to report data from a DB2 subsystem in San Francisco. The location name of the DB2 subsystem is USIBMSTOSQL1.

<table>
<thead>
<tr>
<th>Action</th>
<th>Location</th>
<th>Direction</th>
<th>Time Zone</th>
<th>CPU Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>USIBMSTOSQL1</td>
<td>2 08 00 1 00 00</td>
<td>1 00 00 1 00 00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Figure 136. Time Zone Data 1

The Time Zone column shows the time zone of the location relative to GMT. San Francisco is 8 hours west of Greenwich.

No value is specified in the CPU Clock column to indicate that the CPU clock is set to GMT. If you do not specify a value for the CPU clock, you do not have to change the default direction (1).

In this example a default (*) is specified for locations that do not have an individual entry. If you do not specify a default, no time zone adjustment is made to data from locations that do not have an entry.

If you want to use San Francisco local time in your reporting, specify location USIBMSTOSQL1 as your TIMEZONE location:

```plaintext
GLOBAL
  TIMEZONE (USIBMSTOSQL1)
```

The following adjustments are applied during OMEGamon XE for DB2 PE processing:

- Data from location USIBMSTOSQL1 is adjusted -8 hours because its CPU clock is set to GMT and its local time is eight hours less than GMT.
- Data from locations that do not have an entry is adjusted -8 hours because the default CPU clock setting (*) is GMT as shown in Figure 136.

The reported data reflects San Francisco local time.

The local time is used in FROM/TO processing, printed on reports, and stored in the File and Save data sets.

Example 2: CPU clock settings:
This example demonstrates how to report data from location USIBMSTOSQL1. The only difference to example 1 is that the CPU clock of the location is set to New York local time. This is because the head office of the company is situated in New York.

The value specified in the CPU Clock column shows that the CPU clock of location USIBMSTOSQL1 is set to New York local time, which is five hours west of Greenwich.

A default (*) has been defined for locations that do not have an entry in the data set.

Suppose you again want to use San Francisco local time in your reporting. Specify location USIBMSTOSQL1 as your TIMEZONE location:

```
GLOBAL
    TIMEZONE (USIBMSTOSQL1)
```

Because the time zone for location USIBMSTOSQL1 is eight hours west of Greenwich, the following adjustments are made during Performance Expert processing:

- Data from location USIBMSTOSQL1 is adjusted -3 hours because its CPU clock is set to New York local time, which is 5 hours less than GMT.
- Data from locations that do not have an entry is adjusted -8 hours because the default CPU clock setting (*) is GMT, as shown in Figure 137.

As a result, data from all locations reflects San Francisco local time.

**Example 3: Time zones and CPU clock settings:**

This example shows how to report data from two locations. The first location, USIBMSTOSQL1, is situated in San Francisco, the second, USIBMSTOSQL2, in New York.
The time zone settings for both locations reflect the local time at that site:

- The time zone of location USIBMSTOSQL1 is San Francisco, 8 hours less than the GMT.
- The time zone of location USIBMSTOSQL2 is New York, 5 hours less than the GMT.

The CPU clock of one location is set to GMT and the other to the local time:

- The CPU clock of location USIBMSTOSQL1 is set to zero to indicate GMT.
- The CPU clock of location USIBMSTOSQL2 is set to five to indicate the New York local time.

Again, if you want to use San Francisco local time in your reporting, specify location USIBMSTOSQL1 as your TIMEZONE location:

```
GLOBAL
  TIMEZONE (USIBMSTOSQL1)
```

The following adjustments are made during Performance Expert processing:

- Data from location USIBMSTOSQL1 is adjusted -8 hours because its CPU clock is set to GMT.
- Data from location USIBMSTOSQL2 is adjusted -3 hours because its CPU clock is set to New York local time, which is 5 hours less than GMT.
- Data from locations that do not have an entry is adjusted -8 hours because the default CPU clock setting (*) is GMT, as shown in Figure 138.

As a result, data from all locations reflects San Francisco local time.

For more information about time zone processing, see the description of GLOBAL, and the appendixes in the Report Command Reference.

**Correlation ID translation**

The correlation ID is a DB2 field that identifies the task executed by DB2.

The correlation ID contains:

*Table 14. Contents of correlation identifiers*

<table>
<thead>
<tr>
<th>For batch jobs</th>
<th>Jobname</th>
</tr>
</thead>
</table>

---

**Correlation ID translation**

The correlation ID is a DB2 field that identifies the task executed by DB2.

The correlation ID contains:

*Table 14. Contents of correlation identifiers*

<table>
<thead>
<tr>
<th>For batch jobs</th>
<th>Jobname</th>
</tr>
</thead>
</table>

---

**Correlation ID translation**

The correlation ID is a DB2 field that identifies the task executed by DB2.

The correlation ID contains:

*Table 14. Contents of correlation identifiers*

<table>
<thead>
<tr>
<th>For batch jobs</th>
<th>Jobname</th>
</tr>
</thead>
</table>

---

**Correlation ID translation**

The correlation ID is a DB2 field that identifies the task executed by DB2.

The correlation ID contains:

*Table 14. Contents of correlation identifiers*

<table>
<thead>
<tr>
<th>For batch jobs</th>
<th>Jobname</th>
</tr>
</thead>
</table>

---

**Correlation ID translation**

The correlation ID is a DB2 field that identifies the task executed by DB2.

The correlation ID contains:

*Table 14. Contents of correlation identifiers*

<table>
<thead>
<tr>
<th>For batch jobs</th>
<th>Jobname</th>
</tr>
</thead>
</table>
Table 14. Contents of correlation identifiers (continued)

<table>
<thead>
<tr>
<th>For TSO applications</th>
<th>Original authorization ID (the logon user ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>For applications using the</td>
<td>Original authorization ID (the logon user ID)</td>
</tr>
<tr>
<td>DB2 call attachment facility</td>
<td></td>
</tr>
<tr>
<td>For CICS transactions</td>
<td>Connection type, thread type, thread number, and the transaction ID</td>
</tr>
<tr>
<td>For IMS applications</td>
<td>PST number and PSBNAME of the application</td>
</tr>
</tbody>
</table>

Particularly for CICS and IMS it is useful to break the correlation ID into several parts, so that you can easily distinguish the transaction ID (for CICS threads) from the PSBNAME (for IMS threads).

**The default translation**

OMEGAMON XE for DB2 PE breaks the correlation ID into parts by translating the correlation ID into two separate identifiers, the **correlation name** and the **correlation number**.

Unless it was changed in your installation, this translation is based on the connection type of the thread and is done as follows:

Table 15. The 12-Byte Correlation ID field and the default translation

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Correlation number: blank</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSO, DB2 call attach</td>
<td>Correlation name: original authorization ID</td>
<td>Correlation number: blank</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CICS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Correlation number: pool thread</td>
<td>Correlation name: transaction ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMS</td>
<td>Correlation number: application PSBNAME</td>
<td>Correlation name: application PST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRS</td>
<td>Correlation name: the first 8 characters of the correlation ID provided by the application during signon</td>
<td>Correlation number: the remaining 4 characters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The correlation name and correlation number can be used as OMEGAMON XE for DB2 PE identifiers CORRNAME and CORRNMBR in all report sets.

You can use CORRNAME and CORRNMBR to order data on reports and to include and exclude input records.

**Changing the default translation**

You can override this default translation using the IRF option for maintaining parameter data sets (option 4 on the DB2 PM main menu).

The correlation translation information is kept in the CORRDATA member of the DPMPARMS data set.

Each record in the CORRDATA member specifies the translation that is to be used for a specific connection ID. The connection ID is used here, not the connection type.
The translation is expressed as:
- Offset where the correlation name starts
- Length of the correlation name
- Offset where the correlation number starts
- Length of the correlation number

If OMEGAMON XE for DB2 PE does not find the connection ID for a given thread in the CORRDATA member, the default translation is used.

To activate the tailored correlation translation, specify the DPMPARMS data set for the DPMPARMS ddname.

**Defining the MAINPACK identifier**

In the Accounting report set you can use the MAINPACK identifier to distinguish plans according to the packages they contain. The representative package is either the first or the last package or DBRM executed within a plan.

This identifier is useful when the name of a plan does not provide satisfactory identification, as is the case with DBATs initiated by non-DB2 requesters that all have the same plan name DISTSERV.

You can define certain aspects of the MAINPACK identifier:
- Whether the first or the last package executed within a plan is used as the MAINPACK.
- Whether you want to use the package ID, the collection ID, or the location name of the package name as the value for the identifier. In the case of a DBRM, the program name is always used.

If you wish, you can have different MAINPACK definitions for data from different environments and from different plans.

The MAINPACK identifier can be used to include, exclude, and order data.

When you include data using MAINPACK, data from other packages belonging to the same plan is also reported. (If you used the PACKAGE identifier instead, data for that package, regardless of the plan, would be reported.) The same applies to ordering data.

The default definition for MAINPACK is to use the package ID of the first executed package.

The MAINPACK definition is stored in the member MAINPACK of the DPMPARMS data set. You can access the member using the MAINPACK Definition Member Editor, which is part of the Interactive Report Facility (IRF).

To access the editor, select option 4 (Maintain parameter data sets) from the Performance Expert main menu. The Data Set Maintenance Menu is displayed.
Type 4 in the input field to select Maintain MAINPACK definitions, and type the
name of your DPMPARMS data set on the line below DPMPARMS data set.

Press Enter to go to the MAINPACK Definition Member Editor panel.

Use the editor to enter the main package specifications. Specify the requesting
locations, connection IDs, plan names, and codes for the MAINPACK definition.

**Figure 139. Selecting Maintain MAINPACK Definitions**

Enter the required information in the input fields. You can modify existing
information by typing over it.

**Figure 140. MAINPACK Definition Member Editor panel**

Action

Standard ISPF editor line commands. For example:

I Generate a new line
Delete a line
Duplicate a line
Move a line

Requesting Location
The 16-byte requesting location name. DB2 uses this name to identify the requester DB2 subsystem for distributed threads. For nondistributed threads this is the same as the local location name.

Connection ID
The 8-byte connection name used by DB2 to identify your environment.

Plan Name
The 8-byte name of the DB2 plan. In most cases it is a user-specified name, but for non-DB2 requesters it has a constant value DISTSERV.

Code
One of the predefined definitions of MAINPACK.

You can get information about an entry field by moving the cursor to the field and pressing F1 (Help).

Specifying the MAINPACK (example)
The following example shows how you can define different MAINPACK specifications for different plans. In this example, different MAINPACK specifications for plans DISTSERV and CICSA are defined.

DISTSERV is used as the plan name for all DBATs initiated by non-DB2 requesters. Therefore it can be useful to be able to distinguish between the different DISTSERV plans according to the packages they contain.

The plan CICSA is a large plan consisting of several packages and it is used for many different kinds of transactions. The goal is to distinguish between the different executions of this plan.

The following figure shows the MAINPACK Definition Member Editor panel with the specifications.
In this case, the default entry is for packages that do not have a specific entry. For these plans, the collection ID of the last executed package is used as the MAINPACK.

For plan DISTSERV, the representative package was defined as the first executed package in this example. This was done because it is likely that for the DBATs initiated by non-DB2 requesters the first package usually provides the necessary information to identify the plan. The assumption for this plan was that the package identifier was the most convenient identifier value.

For plan CICSA, the representative package was defined as the last executed package. The reason for doing this was that for this particular plan, in this example, the last executed package best identifies the transaction. The package ID was used as the value of the identifier.

**Examples of using the MAINPACK and PACKAGE identifiers**

The following examples show how you can use the MAINPACK and PACKAGE identifiers. The first example is a short report and shows the short format of the package information. The second example shows the long format of the package information. Both examples show how to use these identifiers in INCLUDE/EXCLUDE and ORDER processing.

**Example 1: Using MAINPACK**

Assume that, during the reporting interval used in this example, two transactions are performed on behalf of a non-DB2 requester. One transaction executes package CUSTINQR and the other transaction executes package ACCTCLOS. You only want to report the activities that are performed by these non-DB2 requesters, that is, DISTSERV plans.

If you use PLANNAME to order the report, you cannot distinguish between the two packages or the two transactions because they both execute plan DISTSERV. So to identify the transactions by means of a package, the MAINPACK identifier is used in the ORDER. For this case the default specification for MAINPACK is used, which is...
to select the package ID of the first package executed within a plan. The Accounting report is then summarized and ordered by this MAINPACK (first package ID).

The following command was used:

```
ACCOUNTING REPORT
   INCLUDE (PLANNAME(DISTSERV))
   ORDER (MAINPACK)
```

The Accounting report looks like this:

```
Figure 142. Short Accounting report - ordered by MAINPACK
```

Notice that the values for the package in the Package Information block represent the processing specific to that package. In this case, there is only one package executed for each transaction. If there had been a second package within the same transaction, then the second package would have been listed as a second line within the Package Information block.

**Example 2: Using PACKAGE**

Assume that, during the reporting interval used in this example, three transactions are performed and that DB2 Accounting Trace classes 7 and 8 are active.

- The first transaction executes plan PLANINQ1 and uses packages CUSTINQA and CUSTINQB and DBRM CUSTINQC.
- The second transaction executes plan PLANINQ2 and uses packages CUSTINQB and CUSTINQD.
- The third transaction executes plan PLANINQ1 also but uses package CUSTINQA only.

All the packages are executed once within the transaction and the transactions are executed only once.

If you want to find out the resource utilization for each package or DBRM, you ORDER the report by PACKAGE and print only the package or DBRM specific data.
from the Accounting records. In this example, you want detailed information about the packages or DBRM, therefore you request a LONG report.

You have previously determined that you are not interested in package CUSTINQD, so you exclude that package. By default all plan names present in the input data are included in the report.

You specify the following command:

```
ACCOUNTING
REPORT
LAYOUT (LONG)
EXCLUDE (PACKAGE(CUSTINQD))
ORDER (PACKAGE)
```

The Package Information blocks for the different packages on the Accounting report look like this:

```
LOCATION: VTAMA OMEGAMON XE for DB2 Performance Expert (V4R2) PAGE: 1-1
GROUP: N/P ACCOUNTING REPORT - LONG REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P TO: NOT SPECIFIED
SUBSYSTEM: SSDQ ORDER: PACKAGE INTERVAL FROM: 04/29/08 20:18:50.43
DB2 VERSION: V7 SCOPE: MEMBER TO: 05/20/08 17:57:56.76

PACKAGE: 'BLANK'.BOSNACOL.CUSTINQA

<table>
<thead>
<tr>
<th>CUSTINQA</th>
<th>VALUE</th>
<th>TIMES</th>
<th>CUSTINQA</th>
<th>AVERAGE TIME</th>
<th>AVG_EV</th>
<th>TIME/EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
<td>'BLANK'</td>
<td>CPU TIME</td>
<td>0.004540</td>
<td>SYNCHRONOUS I/O</td>
<td>0.012075</td>
<td>1.00 0.012075</td>
</tr>
<tr>
<td>COLLECTION ID</td>
<td>BOSNACOL</td>
<td>AGENT</td>
<td>0.004540</td>
<td>OTHER READ I/O</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>PROGRAM NAME</td>
<td>CUSTINQA</td>
<td>SUSPENSION-CL8</td>
<td>0.012075</td>
<td>SERV.TASK SWITCH</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>OCCURRENCES</td>
<td>2</td>
<td>PAR.TASKS</td>
<td>0.000000</td>
<td>OTHER WRITE I/O</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>SQL STMT - AVERAGE</td>
<td>3.00</td>
<td>NOT ACCOUNTED</td>
<td>0.211530</td>
<td>DRAIN LOCK</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>SQL STMT - TOTAL</td>
<td>6</td>
<td>AVG.DB2 ENTRY/EXIT</td>
<td>6.00</td>
<td>CLAIM RELEASE</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>STOR PROC EXECUTED</td>
<td>20</td>
<td>DB2 ENTRY/EXIT</td>
<td>12</td>
<td>PAGE LATCH</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>UDF EXECUTED</td>
<td>0</td>
<td>CPU SERVICE UNITS</td>
<td>11200</td>
<td>SCHEDULE UDF</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>UDF EXECUTED</td>
<td>0</td>
<td>AGENT</td>
<td>8000</td>
<td>NOTIFY MESSAGES</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>UDF EXECUTED</td>
<td>0</td>
<td>PAR.TASKS</td>
<td>3200</td>
<td>GLOBAL CONTENTION</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>SUCC_AUTH_CHECK</td>
<td>0</td>
<td></td>
<td></td>
<td>TOTAL CL8 SUSPENS.</td>
<td>0.012075</td>
<td>1.00 0.012075</td>
</tr>
</tbody>
</table>

Figure 143. Long Accounting report - ordered by PACKAGE

LOCATION: VTAMA OMEGAMON XE for DB2 Performance Expert (V4R2) PAGE: 1-2
GROUP: N/P ACCOUNTING REPORT - LONG REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P TO: NOT SPECIFIED
SUBSYSTEM: SSDQ ORDER: PACKAGE INTERVAL FROM: 04/29/08 20:18:50.43
DB2 VERSION: V7 SCOPE: MEMBER TO: 05/20/08 17:57:56.76

PACKAGE: 'BLANK'.BOSNACOL.CUSTINQB

<table>
<thead>
<tr>
<th>CUSTINQB</th>
<th>VALUE</th>
<th>TIMES</th>
<th>CUSTINQB</th>
<th>AVERAGE TIME</th>
<th>AVG_EV</th>
<th>TIME/EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
<td>'BLANK'</td>
<td>CPU TIME</td>
<td>0.004247</td>
<td>SYNCHRONOUS I/O</td>
<td>0.027006</td>
<td>1.00 0.027006</td>
</tr>
<tr>
<td>COLLECTION ID</td>
<td>BOSNACOL</td>
<td>AGENT</td>
<td>0.004247</td>
<td>OTHER READ I/O</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>PROGRAM NAME</td>
<td>CUSTINQB</td>
<td>SUSPENSION-CL8</td>
<td>0.027006</td>
<td>SERV.TASK SWITCH</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>OCCURRENCES</td>
<td>2</td>
<td>PAR.TASKS</td>
<td>0.000000</td>
<td>OTHER WRITE I/O</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>SQL STMT - AVERAGE</td>
<td>3.00</td>
<td>NOT ACCOUNTED</td>
<td>0.258774</td>
<td>DRAIN LOCK</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>SQL STMT - TOTAL</td>
<td>6</td>
<td>AVG.DB2 ENTRY/EXIT</td>
<td>6.00</td>
<td>CLAIM RELEASE</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>STOR PROC EXECUTED</td>
<td>20</td>
<td>DB2 ENTRY/EXIT</td>
<td>12</td>
<td>PAGE LATCH</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>UDF EXECUTED</td>
<td>0</td>
<td>CPU SERVICE UNITS</td>
<td>11920</td>
<td>SCHEDULE UDF</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>UDF EXECUTED</td>
<td>0</td>
<td>AGENT</td>
<td>7100</td>
<td>NOTIFY MESSAGES</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>UDF EXECUTED</td>
<td>0</td>
<td>PAR.TASKS</td>
<td>4820</td>
<td>GLOBAL CONTENTION</td>
<td>0.000000</td>
<td>0.00 N/C</td>
</tr>
<tr>
<td>SUCC_AUTH_CHECK</td>
<td>0</td>
<td></td>
<td></td>
<td>TOTAL CL8 SUSPENS.</td>
<td>0.027006</td>
<td>1.00 0.027006</td>
</tr>
</tbody>
</table>

Figure 144. Long Accounting report - ordered by PACKAGE (continued)
On this report:

- Information for package CUSTINQA is derived from the first and the third transactions.
- Information for package CUSTINQB is derived from the first and the second transactions.
- Information for DBRM CUSTINQC is derived from the first transaction.

In the example of the long Accounting report, shown in Figure 143 on page 220, each entry begins on a new page, even though all entries fit on a single page. To print all entries on one page you can create your own layout by modifying the sample layout LONG.

You can do this without changing the actual layout. Simply press F3 in the UTR Block Selection panel to display the UTR Save Layout Details panel (see Figure 133 on page 208). From this panel, you can specify no for the question Is each new record or entry to start on a new page? You can save the layout under a new name and use it when ordering by PACKAGE.

### Modifying exception field descriptors

You can change the descriptive labels for exception fields as they are printed for the exception records in batch reports.

To do this, create a member called EXCHANGE in your DPMPARMS data set using the ISPF editor. For each field that requires a new descriptor, enter a line into this member that contains the field name and the new descriptor, separated by a comma without spaces.

### The Performance Database and the Performance Warehouse

OMEGAMON XE for DB2 PE performance data can be loaded into DB2 tables to build a performance database or a performance warehouse.

You can load performance data into DB2 tables to create a performance database. With a performance database you can:
Perform additional performance evaluations. As an example you can examine the access paths of dynamic SQL statements.

- Collect historic data.
- Use SQL for fast and easy retrieval of data.

The performance data can come from the following data groups:
- Accounting
- Audit
- Locking
- Record traces (IFCID 22, 63, 96 and 125)
- Statistics
- System parameters
- Batch, periodic, and display exceptions

Data can be aggregated or nonaggregated:
- In the case of aggregated data, several records are summarized by specific OMEGAMON XE for DB2 PE identifiers. In a report, each entry represents aggregated data. Use the SAVE subcommand to generate a VSAM data set that contains the aggregated data. When the data has been saved, use the Save-File utility to generate a DB2-loadable data set.
- In the case of nonaggregated data, each record is listed in order of occurrence. In a trace, each entry represents nonaggregated data. Use the FILE subcommand to generate a data set that contains the nonaggregated data.

For batch, periodic, and display exceptions, a DB2-loadable data set is automatically generated for each OMEGAMON XE for DB2 PE execution if EXFILDD1 is defined in your JCL. The generated data set is sequential variable-blocked and can be directly loaded into DB2.

The following figure shows how performance data is formatted and loaded for each data group.

---

**Figure 146. How performance data is formatted and loaded**

The following topics provide additional information:
- "Database structure" on page 223
- "Creating data for the Performance Warehouse" on page 231
Database structure

This section shows the structure of each of the tables in the performance database.

Accounting tables

This section shows the structure of each of the Accounting tables in the performance database.

The generated data set consists of the following records:

- **General**: One row per thread.
- **Group buffer pool**: One row per group buffer pool used.
- **Package data**: One row per package and DBRM executed.
- **DDF data**: One row per remote location participating in distributed activity.
- **Buffer pool data**: One row per buffer pool used.
- **RLF**: One row per resource limit type encountered.

When parallel tasks are performed within a data sharing group, the parallel task records are contained in the record of the originating task, regardless of whether the parallel tasks were performed on the same member as the originating task or on other members of the data sharing group.

**Samples**:  
Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE examples can be used for building the DB2 tables into which Accounting File and Save-File data can be loaded. The DB2 LOAD control statements can be used for loading this data into the DB2 tables.

**Accounting File**:  
The parts for Accounting file data are as follows.
Accounting data can contain all or a subset of available data types, depending on
the specification of the DATATYPE subcommand option of the ACCOUNTING FILE
command. Further, the specified data types can be stored in one or several data
sets in any combination by means of multiple invocations of FILE DATATYPE in a
JCL. To load all Accounting data from several output data sets into a DB2 table, you
can run the following DB2 LOAD control statements concurrently. For more
information, see the Report Command Reference.

Table 16. Parts for Accounting file data

<table>
<thead>
<tr>
<th>Type of data</th>
<th>CREATE TABLE statements</th>
<th>LOAD control statements</th>
<th>Table description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>DGOACFGE</td>
<td>DGOALFGE</td>
<td>DGOABFGE</td>
</tr>
<tr>
<td>Group Buffer Pool</td>
<td>DGOACFGP</td>
<td>DGOALFGP</td>
<td>DGOABFGP</td>
</tr>
<tr>
<td>Buffer Pool</td>
<td>DGOACFBU</td>
<td>DGOALFBU</td>
<td>DGOABFBU</td>
</tr>
<tr>
<td>DDF Records</td>
<td>DGOACFDF</td>
<td>DGOALFDF</td>
<td>DGOABFDF</td>
</tr>
<tr>
<td>Package Records</td>
<td>DGOACFPK</td>
<td>DGOALFPK</td>
<td>DGOABFPK</td>
</tr>
</tbody>
</table>

Examples of SQL queries are in member DGOAQFIL.

Accounting Save-File:

The parts for Accounting Save-File data are as follows.

Table 17. Parts for Accounting Save-File data

<table>
<thead>
<tr>
<th>Type of data</th>
<th>CREATE TABLE statements</th>
<th>LOAD control statements</th>
<th>Table description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>DGOACSGE</td>
<td>DGOALSGE</td>
<td>DGOABSGE</td>
</tr>
<tr>
<td>Group Buffer Pool</td>
<td>DGOACSGP</td>
<td>DGOALSGP</td>
<td>DGOABSGP</td>
</tr>
<tr>
<td>Buffer Pool</td>
<td>DGOACSBU</td>
<td>DGOALSBU</td>
<td>DGOABSBU</td>
</tr>
<tr>
<td>DDF Records</td>
<td>DGOACSDF</td>
<td>DGOALSDF</td>
<td>DGOABSDF</td>
</tr>
<tr>
<td>Package Records</td>
<td>DGOACSPK</td>
<td>DGOALSPK</td>
<td>DGOABSPK</td>
</tr>
<tr>
<td>Resource Limit Facility (RLF) records</td>
<td>DGOACSRF</td>
<td>DGOALSRF</td>
<td>DGOABSRF</td>
</tr>
</tbody>
</table>

Save data sets must be converted to the Save-File layout before they can be
loaded into tables.

Examples of CREATE VIEW statements are supplied in the member DGOAVSAV.

Examples of SQL queries are in member DGOAQSAV.

Audit tables

This section shows the structure of each of the Audit tables in the performance
database.
Authorization Failures
One row for each authorization failure that occurred

Authorization Control
One row per authorization control issued

Authorization Change
One row for each of the following authorization changes or authorization establishments:
- SET CURRENT SQLID
- END OF IDENTIFY
- END OF SIGNON
- INBOUND/OUTBOUND DISTRIBUTED TRANSLATION

Audited DDL Access
One row per CREATE or DROP against an Audited object, or ALTER TABLE which change the AUDIT option against an audited table.

Audited DML at Bind Access
One row for each SQL statement that involves audited objects.

Objects in SQL statement
One row for each object involved in the SQL statements recorded in Audited DML at Bind Access.

Audited DML Access
One row for the first change (write) to an audited object in a unit of work

Audited Utility Access
One row for each object and each phase accessed by a utility execution.

SQL Statement
One row for the text of each SQL statement involved in
- Authorization Failure
- Authorization Control
- Audited DDL Access
- Audited DML at Bind Access

Samples:
The parts for Audit file data are as follows.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.
The CREATE TABLE examples can be used for building the DB2 tables into which the Audit File data sets can be loaded. The DB2 LOAD utility control statements can be used for loading these Audit File output records into DB2 tables.

Table 18. Parts for Audit file data

<table>
<thead>
<tr>
<th>Type of data</th>
<th>CREATE TABLE statements</th>
<th>LOAD control statements</th>
<th>Table description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind</td>
<td>DGOXCBND</td>
<td>DGOXLBND</td>
<td>DGOXBBND</td>
</tr>
<tr>
<td>Auth Change</td>
<td>DGOXCCHG</td>
<td>DGOXLCHG</td>
<td>DGOXBCHG</td>
</tr>
<tr>
<td>Auth Control</td>
<td>DGOXCCNT</td>
<td>DGOXLCNT</td>
<td>DGOXBCNT</td>
</tr>
<tr>
<td>DDL</td>
<td>DGOXCDDL</td>
<td>DGOXLDML</td>
<td>DGOXBDML</td>
</tr>
<tr>
<td>DML</td>
<td>DGOXCDML</td>
<td>DGOXLDML</td>
<td>DGOXBDML</td>
</tr>
<tr>
<td>Auth Fail</td>
<td>DGOXCFAI</td>
<td>DGOXLFAI</td>
<td>DGOXBFAI</td>
</tr>
<tr>
<td>SQL</td>
<td>DGOXCSQL</td>
<td>DGOXLSQL</td>
<td>DGOXBSQL</td>
</tr>
<tr>
<td>Utility</td>
<td>DGOXCUTI</td>
<td>DGOXLUTI</td>
<td>DGOXBUTI</td>
</tr>
</tbody>
</table>

Exceptions table
This section shows the structure of the Exceptions table in the performance database.

Figure 149. The Exception table

The generated table contains a list of Accounting and Statistics exceptions:

Batch exceptions
  One row per exception

Periodic exceptions
  One row per exception

Display exceptions
  One row per exception

Samples:

The parts for Exceptions file data are as follows.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE example can be used for building the DB2 tables into which File data sets can be loaded. The DB2 LOAD utility control statements can be used for loading these File output records into DB2 tables.
The query in DGOEQFIL uses the sample view defined in DGOEVFIL.

Table 19. Parts for Exceptions file data

<table>
<thead>
<tr>
<th>Type of data</th>
<th>CREATE TABLE statements</th>
<th>LOAD control statements</th>
<th>Sample query</th>
<th>Table description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception</td>
<td>DGOECFIL</td>
<td>DGOELFIL</td>
<td>DGOEQFIL</td>
<td>DGOEBFIL</td>
</tr>
</tbody>
</table>

**Locking table**

This section shows the structure of the Locking table in the performance database.

Sample query

The table contains a row for each occurrence of the following events:
- A LOCK, UNLOCK, CHANGE, or QUERY request processed by DB2.
- A request to acquire a claim, change a claim duration, or release a claim.
- A request to release a drain on a claim class.
- A successful lock avoidance.

**Samples:**

The parts for Locking file data are as follows.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE example can be used for building the DB2 tables into which the Locking File data sets can be loaded. The DB2 LOAD utility control statements can be used for loading these Locking File output records into DB2 tables.

Table 20. Parts for Locking file data

<table>
<thead>
<tr>
<th>Type of data</th>
<th>CREATE TABLE statements</th>
<th>LOAD control statements</th>
<th>Table description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind</td>
<td>DGOLCFIL</td>
<td>DGOLLFIL</td>
<td>DGOLBFIL</td>
</tr>
</tbody>
</table>

**Record Trace tables**

This section shows the structure of each of the Record Trace tables in the performance database.
The generated data set contains the following records:

**Access path**
- One row per IFCID 22.

**SQL statement**
- One row per SQL statement text record.

**RID pool**
- One row per multiple index access operation.

**SORT END**
- One row for each sort end event triggered by an SQL query.

**Samples:**

The parts for Record Trace File data are as follows.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE examples can be used for building the DB2 tables into which Record Trace File data sets can be loaded. The DB2 LOAD utility control statements can be used for loading these Record Trace File output records into DB2 tables.

Table 21. Parts for Record Trace File data

<table>
<thead>
<tr>
<th>Type of data</th>
<th>CREATE TABLE statements</th>
<th>LOAD control statements</th>
<th>Sample query</th>
<th>Table description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minibind (IFCID 22)</td>
<td>DGONCFMB</td>
<td>DGONLFMB</td>
<td>DGONQFMB</td>
<td>DGONBFMB</td>
</tr>
<tr>
<td>SQL Statement (IFCID 63)</td>
<td>DGONCFSQ</td>
<td>DGONLFSQ</td>
<td>DGONQFSQ</td>
<td>DGONBFSQ</td>
</tr>
<tr>
<td>SORT END (IFCID 96)</td>
<td>DGONCFSE</td>
<td>DGONLFSE</td>
<td>DGONQFSE</td>
<td>DGONBFSE</td>
</tr>
<tr>
<td>RID POOL (IFCID 125)</td>
<td>DGONCFRP</td>
<td>DGONLFRP</td>
<td>DGONQFRP</td>
<td>DGONBFRP</td>
</tr>
</tbody>
</table>

Figure 151. The Record Trace data set
Statistics tables
This section shows the structure of each of the Statistics tables in the performance database.

![Diagram of Statistics table structure]

The generated table contains the following records:

**General data**
- **File data**
  - One row for each Statistics delta record, containing data from IFCID 1 and 2.
  - A delta record is a set of counters that describes the DB2 activity between two consecutive DB2 Statistics records pairs.
- **Save-File data**
  - One row for each Statistics interval record, containing data from IFCID 1 and 2. A Statistics interval record is a set of counters that describes the DB2 activity within the interval specified by the user.

**Buffer pool data**
- One row per buffer pool active at the start of the corresponding delta record for File data, or interval record for Save-File data.

**Group buffer pool data**
- One row per group buffer pool active at the start of the corresponding delta record for File data, or interval record for Save-File data.

**DDF**
- For each delta record (File data) or interval record (Save-File data), one row per remote location participating in distributed activity using DB2 private protocol and one for all remote locations that used DRDA.

**Buffer pool data set**
- One row for each open data set that has an I/O event rate at least one event per second during the reporting interval.

**Samples:**

The parts for Statistics File and Save-File data are as follows.

Examples of CREATE TABLE, LOAD statements, SQL queries, and descriptions of the statistics tables are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.
The CREATE TABLE examples can be used for building the DB2 tables into which the Statistics File data sets and the converted Statistics Save records can be loaded. The DB2 LOAD utility control statements can be used for loading these Statistics File data sets and converted Statistics Save records into DB2 tables.

**Table 22. Parts for Statistics File and Save-File data**

<table>
<thead>
<tr>
<th>Type of data</th>
<th>CREATE TABLE statements</th>
<th>LOAD control statements</th>
<th>Sample query</th>
<th>Table description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Data</td>
<td>DGOSCGEN</td>
<td>DGOSLGEN</td>
<td>DGOSQGEN</td>
<td>DGOSBGEN</td>
</tr>
<tr>
<td>Buffer Pool</td>
<td>DGOSCBUF</td>
<td>DGOSLBUF</td>
<td>DGOSQBUF</td>
<td>DGOSBBUF</td>
</tr>
<tr>
<td>DDF</td>
<td>DGOSCDFD</td>
<td>DGOSLDDF</td>
<td>DGOSQDDF</td>
<td>DGOSBDDF</td>
</tr>
<tr>
<td>Group Buffer Pool</td>
<td>DGOSCGBP</td>
<td>DGOSLGBP</td>
<td></td>
<td>DGOSBGBP</td>
</tr>
<tr>
<td>Buffer pool data set</td>
<td>DGOSCSET</td>
<td>DGOSLSET</td>
<td></td>
<td>DGOSBSET</td>
</tr>
</tbody>
</table>

The sample query in DGOSQDDDF uses the view defined in DGOSVDDF.

The samples provided are valid for Statistics File and Save-File data.

Save data sets must be converted to the Save-File layout before they can be loaded into tables.

**System parameters**

This section lists the system parameters tables in the performance database.

The Performance Database tables for DB2 system parameters are as follows:

- System parameters from IFCID 106
- System facility parameters from IFCID 106
- Alter buffer pool records from IFCID 201
- Buffer pool attributes from IFCID 202
- Group buffer pool attributes from IFCID 230
- Alter group buffer pool records from IFCID 256

To create system parameter data for the Performance Database, use the SYSPARMS command with the FILE subcommand to produce a data set that is suitable for loading into DB2 tables.

**Samples:**

The parts for System Parameters data are as follows.

Examples of CREATE TABLE, LOAD statements, SQL queries, and descriptions of these tables are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE examples can be used for building the DB2 tables into which the System Parameters File data sets can be loaded. The DB2 LOAD utility control samples can be used for loading these File data sets into DB2 tables.
Creating data for the Performance Warehouse

To create performance data, you need to run the appropriate OMEGAMON XE for DB2 PE command with the FILE or SAVE option. If you use the SAVE option, you must convert the data to the FILE format. You can then use the DB2 LOAD utility to move the data into DB2 Performance Database tables.

If you have installed the DB2 Performance Warehouse Client together with the corresponding host data collector, the following tasks can be simplified and largely automated for statistics data through the use of client-initiated processes:

- Creating performance data
- Loading the data into the Performance Database (Warehouse)
- Maintaining the database

For more information about the Performance Warehouse, see Monitoring Performance from Performance Expert Client.
Chapter 6. Further reference information

The following information provides details about specific themes that are of interest on special occasions.

Using GROUP to improve Save data set performance

If you never require separate report entries for one or more OMEGAMON XE for DB2 PE identifiers, you can use GROUP to reduce the uniqueness of the key in the Save data. This can result in fewer reduced records in the Save data set, a smaller Save data set, and improved performance.

The standard key for Save data sets contains a combination of the applicable OMEGAMON XE for DB2 PE identifiers from DB2 instrumentation records. For example, the key for an Accounting save record contains a number of OMEGAMON XE for DB2 PE identifiers. Because of the uniqueness of the standard key, there can be a very large number of reduced records in a Save data set.

If you never require separate report entries for one or more OMEGAMON XE for DB2 PE identifiers, for example, if you only produce Accounting reports by using ORDER(CONNECT-PRIMAUTH-PLANNAME), you can use groups in INCLUDE REDUCE to obtain the minimum number of records in the Save data set.

The following command stream does not use groups. The resulting Save data set contains reduced records for each unique combination of all of the OMEGAMON XE for DB2 PE identifiers in the key.

```
ACCOUNTING
   REDUCE
   SAVE
```

The following command stream uses groups and minimizes the number of records in the Save data set:

```
ACCOUNTING
   REDUCE
   INCLUDE (CORRNAME (G(ALLCNM)),
            CORRNMBR (G(ALLCNU)),
            ORIGAUTH (G(ALLORI)),
            REQLOC (G(ALLREQ))),
   SAVE
```

The Save data set produced by this command stream contains reduced records for each unique value of the OMEGAMON XE for DB2 PE identifiers that are not grouped, for example, PRIMAUTH and PLANNNAME.

During REDUCE processing, the group name is substituted for the original value of the key entry. For example, all records in the reduced data have a correlation name of ALLCNM.
Because the key now contains the group name (not the original value), you cannot use the original values of any of the grouped OMEGAMON XE for DB2 PE identifiers in REPORT or INCLUDE/EXCLUDE in the same command stream or when you restore the data.

Comparing original authorization IDs with primary authorization IDs

This information explains the difference between original authorization ID and primary authorization ID.

Original authorization ID

During connection to DB2 (either by IDENTIFY or SIGNON), an initial authorization value is passed to the connection exit. This value becomes the original authorization ID.

For IDENTIFY:
- If RACF® is active, this value is the verified user ID.
- If RACF is not active, this value is blank.

For IMS SIGNON:
- If RACF is active, this value is the terminal user ID.
- If RACF is not active, this value is either the LTERM name or the PSB name.

For CICS SIGNON:
- This value is determined by the user-defined CICS resource control table (RCT).
  - The connection (authorization) exit can be either the IBM supplied default or user-written, depending upon whether secondary authorization IDs are used.

Primary authorization ID

The primary authorization ID is the value set by the exit. This value is determined according to the following criteria:
- Whether it is an IDENTIFY or a SIGNON
- Whether RACF is active or inactive
- Whether the exit is IBM supplied or user-written
- Whether secondary IDs are being used

Default values can be any of the following:
- The TSO logon ID
- The value of the USER field on the JOB statement
- A default value specified when you install DB2
- The original (unaltered) value

Note:
- The original authorization ID should be used when you attempt to establish accountability of DB2 activity, because the primary authorization ID can be an ID other than the user (group name, for example).
- If your subsystem uses authorization ID translation for distributed activity, the AUTHID reported for DBATs is the translated value.
- For detailed information about authorization IDs, see the DB2 Administration Guide.
Comparing secondary IDs with SQL ID

DB2 uses two other types of authorization IDs:
- Secondary authorization IDs
- SQL ID

If secondary IDs are used, a user-written authorization exit is also required. A secondary list can contain from 1 to 245 secondary IDs. This list is accessed when you establish the primary authorization ID or the SQL authorization ID.

The primary ID and the SQL ID are set during either IDENTIFY or SIGNON. However, only the SQL ID can be changed after connection by the SET CURRENT SQLID statement.

The SQL ID must be either the primary ID or one of the secondary IDs. It is used for implicit name qualifiers, implicit ownership assignment, and GRANT/REVOKE authorization checking.

For more detailed information about authorization IDs, see the DB2 Administration Guide.

Fields affected by roll up during distributed and parallel tasks

The following table lists fields that are not reported by Accounting reports and traces if roll up during distributed activity and parallel query tasks is active.

See “Distributed activity” on page 122 and “Query parallelism considerations” on page 123 for background information.

Table 24. Fields affected by roll up

<table>
<thead>
<tr>
<th>Field name</th>
<th>Field meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPACAANM</td>
<td>ACTIVITY NAME</td>
</tr>
<tr>
<td>QPACAANM_VAR</td>
<td>ACTIVITY NAME</td>
</tr>
<tr>
<td>QPACARNA</td>
<td>DB2 ENTRY/EXIT - AVG.DB2 ENTRY/EXIT</td>
</tr>
<tr>
<td>QPACASCH</td>
<td>SCHEMA NAME</td>
</tr>
<tr>
<td>QPACASCH_VAR</td>
<td>SCHEMA NAME</td>
</tr>
<tr>
<td>QPACCAMN</td>
<td>STORED PROCEDURE EVENTS</td>
</tr>
<tr>
<td>QPACCASACT</td>
<td>SCHED.PROCEDURE SUSP TIME</td>
</tr>
<tr>
<td>QPACCONT</td>
<td>CONSISTENCY TOKEN</td>
</tr>
<tr>
<td>QPACEJST</td>
<td>ENDING TCB CPU TIME</td>
</tr>
<tr>
<td>QPACSCB</td>
<td>BEGINNING STORE CLOCK TIME</td>
</tr>
<tr>
<td>QPACSCCE</td>
<td>ENDING STORE CLOCK TIME</td>
</tr>
<tr>
<td>QPACSPNS</td>
<td>STORED PROCEDURE EXECUTED</td>
</tr>
<tr>
<td>QPACSQLC</td>
<td>SQL STATEMENTS</td>
</tr>
<tr>
<td>QPACUDNU</td>
<td>UDF EVENTS</td>
</tr>
<tr>
<td>QPACUDST</td>
<td>SCHED.UDF SUSP TIME</td>
</tr>
<tr>
<td>QTXAFLG1</td>
<td>RES LIMIT TYPE</td>
</tr>
<tr>
<td>QTXARLID</td>
<td>RLF TABLE ID</td>
</tr>
<tr>
<td>QWACARNA</td>
<td>DB2 ENTRY/EXIT EVENTS</td>
</tr>
</tbody>
</table>
Table 24. Fields affected by roll up (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Field meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWACNID</td>
<td>NETWORK ID VALUE</td>
</tr>
<tr>
<td>QWACSPCP</td>
<td>STORED PROCEDURE TCB TIME</td>
</tr>
<tr>
<td>QWACTREE</td>
<td>TRIG ELAP TIME UNDER ENCLAVE</td>
</tr>
<tr>
<td>QWACTRTE</td>
<td>TRIG TCB TIME UNDER ENCLAVE</td>
</tr>
<tr>
<td>QXMIAP</td>
<td>RID LIST SUCCESSFUL</td>
</tr>
<tr>
<td>QXNSMIAP</td>
<td>RID LIST NOT USED-NO STORAGE</td>
</tr>
</tbody>
</table>

Using ACCOUNTING REDUCE subcommand options to balance performance and data set space

The subcommand options CALCULATE and OPTIMIZE estimate required ACWORK data set space and control the performance of Accounting report generation jobs that involve data aggregation by means of the REDUCE subcommand. The subcommand options are of interest if the reduction of high volume of Accounting data results in inacceptable performance, difficulties to determine adequate ACWORK space, or even abnormally terminated jobs (abend B37).

OMEGAMON XE for DB2 PE provides several commands for which the REDUCE subcommand and its options can be used to reduce the volume of data before the data is passed to subsequent other subcommands like REPORT and SAVE. When OMEGAMON XE for DB2 PE processes a REDUCE subcommand, it holds its temporary data in a REDUCE data set to relieve virtual storage. This temporary data set is either automatically allocated or, if its placement and size needs to be controlled, explicitly specified as a ccWORK DD statement in your JCL. (cc stands for the report set, respectively the command to which the temporary REDUCE data set is associated, for example, AC for the ACCOUNTING command).

The inherent shortcoming with REDUCE is the difficulty to determine in advance an appropriate size of the temporary data set. The required size depends mostly on the amount and complexity of input data to be reduced and on REDUCE subcommand options like FROM/TO, INCLUDE/EXCLUDE, INTERVAL, and BOUNDARY, which influence the reduction factor. In addition, the size of the temporary data set has an inverse proportional effect on the processing time of a REDUCE step. These shortcomings might result in unacceptable performance of report generation jobs or even unexpected job terminations (B37 abends). These problems are more likely if large amounts of Accounting data are reduced.

Beginning with OMEGAMON XE for DB2 PE version 4.2, the REDUCE subcommand (if used with the ACCOUNTING command) provides the CALCULATE and OPTIMIZE subcommand options. The CALCULATE subcommand option provides information about ACWORK space actually used during a REDUCE step. If insufficient ACWORK space is specified, and the job consequently terminates, CALCULATE provides information about the size required to successfully complete the job in a second run. The OPTIMIZE subcommand option can be used to control the preference between performance and ACWORK space usage.

CALCULATE and OPTIMIZE are optional subcommand options. They can be used alone or in combination and specified in any sequence (logically, CALCULATE should be the last of several REDUCE options because they determine what needs to be calculated). Both options are purely performance- and processing-related and
do not manipulate data in any way. For the latter purpose use REDUCE subcommand options like FROM/TO, INCLUDE/EXCLUDE, INTERVAL, and BOUNDARY.

The following sections describe the concepts and the appropriate uses of the CALCULATE and OPTIMIZE subcommand options. For the complete command syntax and for more information about the ACWORK DD statement, see the Report Command Reference.

**Using ACCOUNTING REDUCE CALCULATE to tackle inappropriate ACWORK size allocations**

This section describes how the use of the CALCULATE subcommand option helps to determine an adequate size for the ACWORK data set with regard to the amount and complexity of the data to be reduced.

By default, OMEGAMON XE for DB2 PE automatically allocates a temporary ACWORK data set to hold output data from the REDUCE subcommand. To control the placement and size of this data set, you can also specify a ddname of ACWORK. However, if the allocated size is too small for the temporary data, REDUCE processing terminates because of insufficient space (abend B37). Thereafter, you need to increase the size of the data set and run the job again, still without knowing an appropriate size.

You can use the CALCULATE subcommand option of REDUCE in the following manner in your OMEGAMON XE for DB2 PE command stream to force OMEGAMON XE for DB2 PE to calculate the appropriate size of ACWORK.

```
ACCOUNTING REDUCE
  Other REDUCE subcommand options
  CALCULATE
  REPORT
  SAVE
```

With CALCULATE in the command stream, OMEGAMON XE for DB2 PE counts the number and lengths of records written to ACWORK during ACCOUNTING REDUCE processing.

- In case ACWORK is sufficiently sized, REDUCE processing finishes normally and an information message in DPMLOG provides details about the calculations.
  
  FPEA0800I THE ALLOCATED SPACE FOR ACWORK WAS SUFFICIENT.
  <V1> BYTES HAVE BEEN WRITTEN TO THE ACWORK DATA SET,
  CONSUMED BY <V2> RECORDS WITH AN AVERAGE LENGTH OF <V3>
  AND A MAXIMUM RECORD LENGTH OF <V4>.

  Subsequent subcommands of ACCOUNTING (here, REPORT and SAVE) will be executed.

  In case you specified the size of ACWORK, you can use the value shown as <V1> and compare it with your specification. You might want to adjust your specification to the same dimension, considering a bonus for variations of the input data.

- In case ACWORK is found to be too small, REDUCE processing stops writing to the data set. However, OMEGAMON XE for DB2 PE continues counting and calculating until all input data to REDUCE is processed. An error message in DPMLOG provides details about the minimum required size of ACWORK. Finally, the job terminates with abend B37.
FPEA0801S INSUFFICIENT SPACE WAS ALLOCATED FOR ACWORK.
COMpletely REDUCE DATA WOULD REQUIRE AN ACWORK OF <V1> BYTES,
CONSUMED BY <V2> RECORDS WITH AN AVERAGE LENGTH OF <V3>
AND A MAXIMUM RECORD LENGTH OF <V4>.

Subsequent subcommands of ACCOUNTING (here, REPORT and SAVE) will not
be executed.

In this case, you need to specify the size of ACWORK to at least the value
shown as <V1> (plus 20% is recommended). Then, you need to rerun the job a
second time.

In both messages the other values, shown as <V2>, <V3>, and <V4>, describe the
amount and quality of processed records. The error message FPEA0801S is
accompanied by other error messages indicating a failure to write to ACWORK. For
more information, see Messages.

Using ACCOUNTING REDUCE OPTIMIZE to balance REDUCE
performance

This section describes how the use of the OPTIMIZE subcommand option and its
parameters helps to control CPU consumption in conjunction with required
ACWORK space.

The REDUCE process takes place in virtual storage and requires several cycles to
complete. By default, without the use of the OPTIMIZE subcommand option, the
amount of virtual storage made available is dynamically adjusted for each cycle,
based on the number of entries to be held in virtual storage for each cycle. This
automatism can result in inappropriate allocation of virtual storage and in poor job
performance.

With OPTIMIZE in use, constant thresholds for all cycles are used for the number of
entries to be held in virtual storage, which results in a constant amount of available
virtual storage. The threshold values are internally derived and are adjusted based
on the OPTIMIZE parameter provided by you.

In summary, a smaller amount of virtual storage holds fewer entries, which results
in faster access, which decreases CPU consumption, but requires more ACWORK
space. Vice versa, a larger amount of virtual storage holds more entries, which
results in slower access, which increases CPU consumption, but requires less
ACWORK space.

You can use the OPTIMIZE subcommand option as outlined below in your
OMEGAMON XE for DB2 PE command stream to control whether the REDUCE
process is performed with minimum CPU consumption (but maximum ACWORK
space), minimum ACWORK space (but maximum CPU consumption), or any
distinct value between these two boundary values.

ACCOUNTING
   REDUCE
     Other REDUCE subcommand options
     OPTIMIZE(parameter)
     REPORT
     SAVE

The OPTIMIZE subcommand option requires one of the following keywords or
values. For more information about other ACCOUNTING subcommands and their
options and interrelations, see the Report Command Reference.
The OPTIMIZE keywords and values have the following effects:

**CPU TIME**
The CPU consumption is minimized, at the expense of required ACWORK space. This setting is equivalent to OPTIMIZE(0).

**ACWORK**
The required ACWORK space is minimized, at the expense of CPU consumption. This setting is equivalent to OPTIMIZE(100).

\( n \) A numeric value from 0 to 100, specifying the balance between minimized CPU consumption (0) and minimized required ACWORK space (100).

The correlation between a specific value and a resource consumption (required ACWORK space) is not exactly linear. The reason is that any value from 0 to 100 that you specify is internally correlated to the maximum number of entries held in virtual storage. Nevertheless, this should be of limited interest for the sake of the intended usage.

The results from different OPTIMIZE settings, assuming identical input data to the REDUCE process, are nearly identical. Small differences should be considered normal. They result from different contents of ACWORK at different stages and from repeated multiplications of rounded values.

**Example of fine-tuning by varying REDUCE OPTIMIZE values**

This example shows one of several methods to fine-tune the data reduction process by monitoring the effects of different OPTIMIZE values for a given amount of data to be reduced. The goal is to find a range of OPTIMIZE values that results in acceptable CPU time and ACWORK space consumption.

To show noticeable variations in CPU time and ACWORK space consumption, it is assumed that the volume of data to be reduced is relatively high and the data is complex. Small trace samples might not show any effects. If your actual data does not result in noticeable variations, there is nothing to optimize.

In Table 25 on page 240 the resulting values for CPU time and ACWORK space consumption are derived from left to right.

1. The job is run with OPTIMIZE(0), which results in CPU time = 10 (acceptable, cannot be improved) and ACWORK = 50 (not acceptable).
   Let's try the other extreme.

2. The job is run with OPTIMIZE(100), which results in CPU time = 35 (not acceptable) and ACWORK = 25 (acceptable, cannot be improved).
   There might exist an OPTIMIZE(n) value between 0 and 100 that satisfies both criteria. Let's try the medium.

3. The job is run with OPTIMIZE(50), which results in CPU time = 25 (not acceptable) and ACWORK = 38 (acceptable).
   OPTIMIZE(50) is still too high; we need to favor CPU time by using an OPTIMIZE(n) value between 0 and 50.

4. The job is run with OPTIMIZE(25), which results in CPU time = 15 (acceptable) and ACWORK = 45 (not acceptable).
OPTIMIZE(25) is too low; we need to favor ACWORK by using an OPTIMIZE(n) value between 25 and 50.

5. The job is run with OPTIMIZE(38), which results in CPU time = 19 (acceptable) and ACWORK = 42 (acceptable).

OPTIMIZE(38) satisfies both criteria. Let’s try to favor ACWORK by using an OPTIMIZE(n) value between 38 and 50.

6. The job is run with OPTIMIZE(44), which results in CPU time = 23 (acceptable) and ACWORK = 41 (acceptable).

OPTIMIZE(44) still satisfies both criteria. Let’s try again to favor ACWORK by using an OPTIMIZE(n) value between 44 and 50.

7. The job is run with OPTIMIZE(47), which results in CPU time = 24 (acceptable) and ACWORK = 39 (acceptable).

OPTIMIZE(47) still satisfies both criteria. Let’s try again to favor ACWORK by using an OPTIMIZE(n) value between 47 and 50.

8. The job is run with OPTIMIZE(49), which results in CPU time = 24 (acceptable) and ACWORK = 39 (acceptable).

OPTIMIZE(49) still satisfies both criteria. In fact nothing has changed.

Based on the given workload and criteria, this example ascertains acceptable OPTIMIZE(n) values from 38 to 49.

<table>
<thead>
<tr>
<th>OPTIMIZE value</th>
<th>0</th>
<th>25</th>
<th>38</th>
<th>44</th>
<th>47</th>
<th>49</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumed CPU time [sec]</td>
<td>10</td>
<td>15</td>
<td>19</td>
<td>23</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Acceptable?</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Consumed ACWORK space [MB]</td>
<td>50</td>
<td>45</td>
<td>41</td>
<td>39</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>Acceptable?</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Both acceptable?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

You will find the actual values for consumed CPU time in the job output of JESMSGLG (search for "TOTAL CPU TIME="). The actual size of ACWORK can be determined through TSO option 3.2, or from messages FPEA0800I and FPEA0801S resulting from REDUCE CALCULATE.

The previous example also reveals the reverse effects of consumed CPU time and ACWORK space for ordered OPTIMIZE values.

<table>
<thead>
<tr>
<th>OPTIMIZE value</th>
<th>0</th>
<th>25</th>
<th>38</th>
<th>44</th>
<th>47</th>
<th>49</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumed CPU time [sec]</td>
<td>10</td>
<td>15</td>
<td>19</td>
<td>23</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>Consumed ACWORK space [MB]</td>
<td>50</td>
<td>45</td>
<td>42</td>
<td>39</td>
<td>39</td>
<td>25</td>
</tr>
</tbody>
</table>

Troubleshooting empty reports

This topic lists and explains several reasons why your report might not contain any report data. Use this information as a guide to identify possible reasons, including inappropriate command parameters.
At its core, OMEGAMON XE for DB2 PE identifies data to be collected and reported by IFCIDs. [DB2 instrumentation data” on page 25] provides an overview of how IFCIDs map to report sets, DB2 trace types and DB2 trace classes. The basic rule is that data to be reported needs to be collected and made available as input data to the report generation process. If required IFCIDs are missing, the generated report ends with NO DATA TO REPORT.

The problem of missing IFCIDs might also be caused by different versions of DB2 or OMEGAMON XE for DB2 PE. Both products might support different sets of IFCIDs, which can lead to situations where an IFCID is either not provided by a DB2 version or cannot be processed by an OMEGAMON XE for DB2 PE version. The same problem might arise if data from other sources, like SMF or GTF, is used for reports.

OMEGAMON XE for DB2 PE provides flexible means to specify which data to collect and which data to report. However, it provides no means to protect against inappropriate use of commands, subcommands and subcommand options, which also can result in empty reports.

The following sections list possible reasons why your report might end with a final message of NO DATA TO REPORT.

Messages indicating unsupported IFCIDs or IFCID formats

If the job summary log shows message FPEC4015I NUMBER OF RECORDS FROM UNSUPPORTED RELEASES OF DB2 WAS ...., OMEGAMON XE for DB2 PE has detected IFCIDs in its input data that it is not supposed to handle. Either the input data is from an outdated DB2 version and the IFCID is no longer supported by OMEGAMON XE for DB2 PE, or the input data contains IFCIDs from a newer DB2 version and OMEGAMON XE for DB2 PE is not yet aware of them.

If the job summary log shows message FPEC4020I NUMBER OF RECORDS FROM UNSUPPORTED PRODUCT RELEASES WAS ...., the input data (such as input data in DPMOUT format) was created by earlier versions of OMEGAMON XE for DB2 PE or OMEGAMON XE for DB2 PM and is no longer supported.

Identifying missing IFCIDs

Report sets such as Accounting, Locking, or Statistics require specific IFCIDs from DB2 instrumentation data, as outlined in [DB2 instrumentation data” on page 25]. If the input data sets (specified with the INPUTDD statement in your job stream) do not contain these IFCIDs, the generated report ends with NO DATA TO REPORT.

For example, an Accounting report requires IFCIDs 3 and 239 in the input data. If the job summary log shows nothing more than in the following example (no IFCIDs 3 and 239), your Accounting report remains empty.

<table>
<thead>
<tr>
<th>IFCID</th>
<th>INPUT COUNT</th>
<th>INPUT PCT OF TOTAL</th>
<th>PROCESSED COUNT</th>
<th>PROCESSED PCT OF TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>89</td>
<td>33.58%</td>
<td>0</td>
<td>0.00%</td>
</tr>
<tr>
<td>2</td>
<td>88</td>
<td>33.20%</td>
<td>0</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Note: The job summary log gives a clear indication about which IFCIDs are contained in your input data. If an IFCID is not listed in the IFCID column or the INPUT COUNT column shows a count of 0, your input data does not contain this IFCID.
Reasons for missing IFCIDs

IFCIDs might be missing in your reports because they are either not collected (missing in the input data for the report generation) or being accidentally suppressed or filtered by inappropriate use of commands, subcommands, or subcommand options.

To identify why specific IFCIDs are not collected, you need to review the methods of how the data was collected. [Collecting data for reports on page 11] describes several methods, some of them let you explicitly specify what to collect. For example, the DB2 startup parameters might determine that certain trace classes (with their implied IFCIDs) are not started.

If your input data to specific reports contains the required IFCIDs, but your reports still do not show any data, it is likely that restrictive command or subcommand options are in use in your job stream.

- The GLOBAL command might apply global filters to all data serving as input to subsequent OMEGAMON XE for DB2 PE commands like ACCOUNTING or STATISTICS.
- OMEGAMON XE for DB2 PE commands like ACCOUNTING or STATISTICS might use subcommand options that further restrict the data. The most important ones are:
  - FROM/TO might limit the time frame too much.
  - INCLUDE/EXCLUDE might limit a DB2 trace class or an IFCID.

For more information about commands and subcommand options, see the Report Command Reference.

For example, a Statistics report requires IFCIDs 1 and 2. The following job summary log shows that 48 records of each IFCID are included in the input data. However, none of these are being processed (PROCESSED COUNT equals 0) during the report generation and consequently the Statistics report is empty.

<table>
<thead>
<tr>
<th>IFCID</th>
<th>INPUT COUNT</th>
<th>INPUT PCT OF TOTAL</th>
<th>PROCESSED COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>48</td>
<td>25.00%</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
<td>25.00%</td>
<td>0</td>
</tr>
</tbody>
</table>

Insufficient amount of Statistics record pairs (IFCIDs 1 and 2)

Output in a Statistics report requires at least two DB2 Statistics record pairs (IFCIDs 1 and 2) in the input data. These IFCIDs are collected at a configurable interval (DB2 system parameter STATIME). For more information, see [Statistics report set on page 160].

An empty Statistics report usually indicates that either the input data does not cover at least two intervals or that filters restrict the amount of data being processed during report generation.

The following example of a job summary log shows that a sufficient amount of IFCID 1 and 2 records are in the input data (INPUT COUNT column). However, filters have limited the number of processed records below the required minimum of two record pairs (PROCESSED COUNT column). Consequently, the Statistics report is empty.
The following job stream example shows how improper use of subcommand options cause an empty Statistics report. Here, the REDUCE step causes a summarization of input data to an interval of 60 minutes, with a boundary to the start of the hour. However, in the REPORT step the FROM and TO subcommand options limit the data to 30 minutes, which is less than a single interval.

```
STATISTICS
  REDUCE
    INTERVAL(60)
    BOUNDARY(60)
  REPORT
    FROM (12/22/08,12:15:00.00)
    TO (12/22/08,12:45:00.00)
```

**Messages indicating incomplete Accounting data (IFCIDs 3 and 239)**

If the job summary log shows one or more of the following messages, incomplete Accounting data was found in the input data and your Accounting report might be incomplete. In the worst case, the report might end with NO DATA TO REPORT.

- FPEA4531I GENERAL ACCOUNTING DATA FOR LUWID INSTANCE V1 IS MISSING. IFCID 3 IS MISSING.
- FPEA4532I ACCOUNTING DATA FOR A NUMBER OF PACKAGES FOR LUWID V1 IS MISSING. ONE OR MORE IFCID 239 IS MISSING.
- FPEA4534I COORDINATING PARALLEL TASK FOR LUW INSTANCE V1 IS NOT REPORTED AS NOT ALL INFORMATION FOR ASSISTING PARALLEL TASKS FROM MEMBER V1 IS AVAILABLE.

Accounting reports require IFCID 3 for reporting a thread. However, package or program data belonging to a thread is contained in multiple 239 IFCIDs. A 1:n relation is established within the data. For correct reporting a complete set of these IFCIDs is required in the input data.

In case of DB2 Query Parallelism, additional sets of IFCIDs 3 and 239 are required. They were created from parallel processors or from DB2 systems that assisted an initiating agent. Dependent on the degree of parallelism, this can result in a 1:p relation between agent and parallel tasks. In other words, correct Accounting processing needs from the agent one IFCID 3 and n IFCIDs 239, and for each of the p parallel tasks also one IFCID 3 and a certain number of IFCIDs 239.

Accounting processing needs a complete set of these IFCIDs for a correct reporting. If either IFCIDs 239 belonging to an IFCID 3 are missing, or if IFCIDs 239 are available but their IFCID 3 is missing, the entire thread is excluded from further processing and reporting.

### Accessibility

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use software products successfully.
This index describes the major accessibility features in OMEGAMON XE for DB2 PE, OMEGAMON XE for DB2 PM, and Buffer Pool Analyzer:

- You can operate all features using the keyboard instead of the mouse.
- You can change the system settings for high contrast for all user interface controls and client area contents.

The following sections explain how to use these accessibility features.

**Operating all features by using the keyboard**

You can use keys or key combinations to perform operations that can also be done through mouse actions. All menu items can be accessed from the keyboard. In those cases, the keyboard equivalent appears to the right of the menu item, or the shortcut letter is underlined. Some keyboard items also have shortcuts.

To navigate through a window or dialog by using the keyboard instead of the mouse, use the following keyboard shortcuts:

<table>
<thead>
<tr>
<th>Navigating through a window or dialog</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access the menu bar in a window</td>
<td>Alt+underlined letter or F10</td>
</tr>
<tr>
<td>Activate a menu item in the menu bar</td>
<td>Enter</td>
</tr>
<tr>
<td>Access controls in a dialog</td>
<td>Alt+underlined letter</td>
</tr>
<tr>
<td>Navigate through the menu bar</td>
<td>Right arrow, left arrow, down arrow, up arrow</td>
</tr>
<tr>
<td>Move to the next set of controls</td>
<td>Tab or Ctrl+Tab</td>
</tr>
<tr>
<td>Move to the previous set of controls</td>
<td>Shift-Tab</td>
</tr>
<tr>
<td>Move within tables</td>
<td>Tab or right arrow, Shift-Tab or left arrow, down arrow, up arrow</td>
</tr>
<tr>
<td>Move within trees</td>
<td>Up arrow, down arrow</td>
</tr>
<tr>
<td>Expand a tree node</td>
<td>Right arrow</td>
</tr>
<tr>
<td>Collapse a tree node</td>
<td>Left arrow</td>
</tr>
<tr>
<td>Move within list boxes</td>
<td>Up arrow, down arrow</td>
</tr>
<tr>
<td>Move within combo boxes</td>
<td>Up arrow, down arrow</td>
</tr>
<tr>
<td>Move to a split bar in a window</td>
<td>F8</td>
</tr>
<tr>
<td>Resize a window by using the split bar</td>
<td>Arrow keys</td>
</tr>
</tbody>
</table>

To perform main tasks in windows by using the keyboard instead of the mouse, use the following keyboard shortcuts:

<table>
<thead>
<tr>
<th>Task</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Help</td>
<td>F1</td>
</tr>
<tr>
<td>Close the active window</td>
<td>Ctrl+W or Alt+F4</td>
</tr>
<tr>
<td>Exit the application</td>
<td>Ctrl+Q</td>
</tr>
<tr>
<td>Cut</td>
<td>Ctrl+X</td>
</tr>
<tr>
<td>Copy</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Paste</td>
<td>Ctrl+V</td>
</tr>
</tbody>
</table>
Table 28. Keyboard shortcuts in a window (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>Ctrl+P</td>
</tr>
<tr>
<td>Refresh displayed data</td>
<td>F5</td>
</tr>
<tr>
<td>Enter or leave history mode</td>
<td>Ctrl+H</td>
</tr>
<tr>
<td>Move back in history mode</td>
<td>Alt+left arrow</td>
</tr>
<tr>
<td>Move forward in history mode</td>
<td>Alt+right arrow</td>
</tr>
<tr>
<td>Add a new subsystem in System Overview</td>
<td>Ctrl+N</td>
</tr>
<tr>
<td>Add a new configuration in Trace Configurations</td>
<td>Ctrl+N</td>
</tr>
<tr>
<td>Open a report in Buffer Pool Analyzer</td>
<td>Ctrl+O</td>
</tr>
<tr>
<td>Start a simulation process in Buffer Pool Analyzer</td>
<td>Ctrl+I</td>
</tr>
<tr>
<td>Cancel Thread in Thread Summary and Details</td>
<td>Del</td>
</tr>
<tr>
<td>Delete Trace Configuration in Trace Configurations</td>
<td>Del</td>
</tr>
<tr>
<td>Open Trace Configuration</td>
<td>Ctrl+Alt+N</td>
</tr>
<tr>
<td>Open Trace Activation</td>
<td>Ctrl+Alt+A</td>
</tr>
<tr>
<td>Open Exception Processing</td>
<td>Ctrl+Alt+E</td>
</tr>
<tr>
<td>Open System Overview</td>
<td>Ctrl+Alt+O</td>
</tr>
<tr>
<td>Open Buffer Pool Analyzer</td>
<td>Ctrl+Alt+B</td>
</tr>
<tr>
<td>Open DB2 command for the active subsystem</td>
<td>Ctrl+Alt+D</td>
</tr>
<tr>
<td>Open Statistics Details for the active subsystem</td>
<td>Ctrl+Alt+S</td>
</tr>
<tr>
<td>Open System Health for the active subsystem</td>
<td>Ctrl+Alt+H</td>
</tr>
<tr>
<td>Open All Threads in Lock Conflict for the active subsystem</td>
<td>Ctrl+Alt+L</td>
</tr>
<tr>
<td>Open Locking Conflicts for the active subsystem</td>
<td>Ctrl+Alt+C</td>
</tr>
<tr>
<td>Open System Parameters for the active subsystem</td>
<td>Ctrl+Alt+Y</td>
</tr>
<tr>
<td>Open Performance Warehouse for the active subsystem</td>
<td>Ctrl+Alt+P</td>
</tr>
<tr>
<td>Open Thread Summary for the active subsystem</td>
<td>Ctrl+Alt+T</td>
</tr>
</tbody>
</table>

To perform main tasks in dialogs by using the keyboard instead of the mouse, use the following keyboard shortcuts:

Table 29. Keyboard shortcuts in a dialog

<table>
<thead>
<tr>
<th>Task</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confirm a dialog</td>
<td>Enter</td>
</tr>
<tr>
<td>Cancel a dialog</td>
<td>Esc</td>
</tr>
<tr>
<td>Activate a button that has the focus</td>
<td>Spacebar or Enter</td>
</tr>
<tr>
<td>Select and deselect check boxes and radio buttons</td>
<td>Spacebar</td>
</tr>
<tr>
<td>Navigate within check box and radio button groups</td>
<td>Arrow keys</td>
</tr>
<tr>
<td>Open combination box menu</td>
<td>Alt+down arrows</td>
</tr>
<tr>
<td>Close combination box menu</td>
<td>Esc</td>
</tr>
</tbody>
</table>
Table 29. Keyboard shortcuts in a dialog (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move up and down in combination box menu</td>
<td>Up arrow, down arrow</td>
</tr>
<tr>
<td>Activate combination box menu item</td>
<td>Spacebar or Enter</td>
</tr>
<tr>
<td>Move within a list box</td>
<td>Up arrow, down arrow</td>
</tr>
<tr>
<td>Activate a list box entry</td>
<td>Enter</td>
</tr>
<tr>
<td>Move between the pages of a window that</td>
<td>Right arrow, left arrow</td>
</tr>
<tr>
<td>contains tabs if the tab has the focus</td>
<td></td>
</tr>
</tbody>
</table>

Changing the system settings for high contrast and font size

You can change the system settings for high contrast and font size.

To use specific settings that are designed for easy reading, take these steps:
1. In the OMEGAMON XE for DB2 PE - System Overview window, click Monitor → Configuration.
2. In the Configuration window, click Accessibility.
3. To specify the appearance scheme for high contrast and font size, select one of the following options:
   - To use the default settings, select [None].
   - To specify a white background containing black text in normal font size, select Black on white.
   - To specify a white background containing black text in large font size, select Black on white (large font).
   - To specify a black background containing white text in normal font size, select White on black.
   - To specify a black background containing white text in large font size, select White on black (large font).

You need to restart the application before the new settings come into effect.
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland GmbH
Department 0790
Pascalstrasse 100
70569 Stuttgart
Germany
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information”


Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.
Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
Bibliography

IBM Tivoli OMEGAMON XE for DB2 Performance Expert publications

IBM Tivoli OMEGAMON XE for DB2 Performance Expert and Performance Monitor on z/OS, Version 4 Release 2 product information
- Configuration and Customization, GC19-2511
- Monitoring Performance from ISPF, SC19-2509
- Monitoring Performance from the OMEGAMON Classic Interface, SC19-2507
- Monitoring Performance from Performance Expert Client, SC19-2508
- Report Command Reference, SC19-2505
- Report Reference, SC19-2504
- Reporting User’s Guide, SC19-2510
- Messages, GC19-2506
- Program Directory for Performance Monitor, GI10-8723
- Program Directory for Performance Expert, GI10-8721

IBM DB2 Buffer Pool Analyzer for z/OS, Version 4 Release 2 product information
- Buffer Pool Analyzer Configuration Guide, SC19-2513
- Program Directory for IBM DB2 Buffer Pool Analyzer for z/OS, GI10-8725

IBM DB2 Performance Expert for z/OS, Version 2 Release 1 and IBM DB2 Performance Monitor for z/OS, Version 8 Release 1 product information
- Monitoring Performance from ISPF, SC18-7975
- Monitoring Performance from the Workstation, SC18-7976
- Report Command Reference, SC18-7977
- Report Reference, SC18-7978
- Reporting User’s Guide, SC18-7979
- Program Directory for IBM DB2 Performance Expert for z/OS, GI10-8549
- Program Directory for IBM DB2 Performance Monitor for z/OS, GI10-8550

IBM Tivoli Monitoring publications

IBM Tivoli Monitoring, Version 6 Release 2 product information
- Installation and Setup Guide, GC32-9407
- Administrator’s Guide, SC32-9408
- User’s Guide, SC32-9409
- Configuring the Tivoli Enterprise Monitoring Server on z/OS, SC27-2313

IBM DB2 publications

IBM DB2 for z/OS, Version 9 Release 1 product information
- Administration Guide, SC18-9840
- Application Programming and SQL Guide, SC18-9841
- Application Programming Guide and Reference for Java, SC18-9842
- Codes, GC18-9843
- Command Reference, SC18-9844
- Data Sharing: Planning and Administration, SC18-9845
- Diagnosis Guide and Reference, LY37-3218
- Diagnostic Quick Reference, LY37-3219
- Installation Guide, GC18-9846
- Internationalization Guide, SC19-1161
• Introduction to DB2 for z/OS, SC18-9847
• Licensed Program Specifications, GC18-9848
• Messages, GC18-9849
• ODBC Guide and Reference, SC18-9850
• Performance Monitoring and Tuning Guide, SC18-9851
• RACF Access Control Module Guide, SC18-9852
• Reference for Remote DRDA Requesters and Servers, SC18-9853
• Reference Summary, SX26-3854
• SQL Reference, SC18-9854
• Utility Guide and Reference, SC18-9855
• What's New?, GC18-9856
• XML Extender Administration and Programming, SC18-9857
• XML Guide, SC18-9858

IBM DB2 Universal Database for z/OS publications

DB2 Universal Database for z/OS, Version 8 product information

• DB2 Administration Guide, SC18-7413
• DB2 Application Programming and SQL Guide, SC18-7415
• DB2 Application Programming Guide and Reference for Java, SC18-7414
• DB2 Command Reference, SC18-7416
• DB2 Data Sharing: Planning and Administration, SC18-7417
• DB2 Diagnosis Guide and Reference, LY37-3201
• DB2 Diagnostic Quick Reference Card, LY37-3202
• DB2 Installation Guide, GC18-7418
• DB2 Licensed Program Specifications, GC18-7420
• DB2 Messages, GC18-8602
• DB2 Codes, GC18-8603
• DB2 ODBC Guide and Reference, SC18-7423
• DB2 Reference Summary, SX26-3853
• DB2 Release Planning Guide, SC18-7425
• DB2 SQL Reference, SC18-7426
• DB2 Utility Guide and Reference, SC18-7427
• DB2 What’s New?, GC18-7428
• DB2 XML Extender for z/OS Administration and Programming, SC18-7431
• Program Directory for IBM DB2 Universal Database for z/OS, GI10-8566

Other IBM publications

• Tivoli OMEGAMON XE and Tivoli Management Services on z/OS: Common Planning and Configuration Guide, SC23-9734
• DFSMS/MVS: Access Method Services for ICF, SC26-4906
• DFSMS/MVS: Access Method Services for VSAM Catalogs, SC26-4905
• DFSMS/MVS: Macro Instructions for Data Sets, SC26-4913
• OS/390 ISPF: Dialog Developer’s Guide and Reference, SC28-1273
• OS/390 ISPF: Services Guide, SC28-1272
• OS/390 MVS: Initialization and Tuning Guide, SC28-1751
• OS/390 MVS: Programming: Authorized Assembler Services Guide, GC28-1763
• OS/390 MVS: Programming: Authorized Assembler Services Reference, GC28-1764 to GC28-1767
• OS/390 MVS: System Codes, GC28-1780
• OS/390 MVS: System Management Facilities (SMF), GC28-1783
• z/OS TSO/E: Messages, SA22-7786
• z/OS DFSORT Application Programming Guide, SC26-7523
• *OS/390 TSO/E: REXX Reference, SC28-1975*
• *TCP/IP: Tutorial and Technical Overview, GG24-3376*

### How to order publications

For information about how to order IBM publications, see "Ordering publications" on page xi.
Index

A
access path 75
accessibility xii, 244
Accounting 52
  class 1 times 127
class 2 times 127
DB2 trace 25
description 6
distributed activity 122
distributed data 183
distributed times 125
effective reports 129, 134, 135, 143, 146, 157
exception processing 134
exception report 64
general 120
grope-scope report 190
input for reports and traces 127
MAINPACK identifier 215, 221
missing data 129
nondistributed times 125
ORDER processing 135
processing considerations 129
query parallelism 123
record generation 128
tailoring report and trace layouts 198, 208
TOP processing 143
trace classes 124
when records are written 128
where information comes from 124
Accounting classes 128
Accounting report
  rolled up values 123, 124
Accounting report (short)
  generating
    with ISPF/PDF 35
tailoring 189
Accounting report set 119
Accounting trace
  rolled up values 123, 124
ACCUMACC
  subsystem parameter 123
accumulated value 161
Accumulated value (delta) 162
Accumulated value (interval) 163
aggregation 120
allied thread 121
allied-distributed thread 121
definition 184
audit
  description 7
  trace 21
Audit trace
  starting automatically 21
AUTHID/PRIMAUTH identifier
  comparing original with primary 234, 235
DB2 START TRACE command 23
authorization
  explain 78
authorization ID
  See AUTHID/PRIMAUTH identifier
auxiliary command
  See also command
  See also subcommand
description 33

B
batch processing
  description 4
batch report
  creating 31
batch reporting 1
bibliography 251
block of data
  adding 202
  replacing fields 204
Block Selection panel 201
buffer pool efficiency 49

C
CALCULATE subcommand option of REDUCE 236
CASE command 33
class 3 suspensions
  server 127
CLASS option 23
class, trace 25
description 25
classes, Accounting 128
classes, Accounting trace 124
classes, trace 128
Collect Report Data (CRD) panels
  access to 13
Collect Report Data Batch
  program 28
Collect Report Data Batch program
  syntax 28
Collect Report Data facility
  collecting data for batch processing 12
  DB2 START TRACE command 21, 25
INSTALL parameter values 21
  using IFCIDs 25
Collect Report Data panel 13
collect task
  configuring 14
  restricting data types 16
  restricting identifiers 17
  restricting IFCIDs 16
  starting and stopping traces 18
collecting data for batch processing 12
collecting data with Near-Term History Data Collector 30
The document contains various sections on command, accounting, data definition name (ddname), data set, data set maintenance menu, correlation number, correlation name, correlation identifier, cor name, correlation identifier, default translation, correlation number, default translation, correlation translation, defaults, cor name identifier, default translation, cor number identifier, default translation, CP parallelism, CPU clock setting, example, CRD (Collect Report Data) panels, access to, customizing, delta record, DEST option, DGOBFLD3 panel, DGOBFLD4 panel, DGOBMENU panel, DGOBPBLK panel, DGOBPLAY panel, DGOBPSAV panel, DGOETV41, different versions of, sample Exception Threshold data set member, warning message, DGOFEP00 panel, DGOMAP00 panel, DGOMAP10 window, DGOMAP30 window, DGOMAP31t window, DGOMAP32 window, DGOMAP33 window, DGOMAP40 window, DGOOJOBM panel, DGOPMENU panel, defining exception thresholds.
DGOPMENU panel (continued)
  defining the MAINPACK identifier 215
  specifying time zones 209
DGOPPLDS panel 210
DGOPPMDM panel 216
DGOPXDS1 panel 59
DGOPXDS2 panel 60
DGOPXDSN panel 61, 65
disability xii, 244
DI/STDD data set 32
  distributed Accounting times 125
  distributed activity 122
  effects on roll up 123
Distributed Data Facility (DDF)
  See DDF (Distributed Data Facility)
  distributed transaction 122
DPMLOG data set 32
DPMLOG execution log
  description 5
DPMOUT data set
  description 32
  streamlining processing 197
DPMPARMS data set 32

Exception Threshold data set (continued)
  block size 178
  different versions of 177
  modifying 177
  purpose 177
  record format 178
  record length 178
  sample 66
  sample member 65
  sample RKO2DATA(DGOETV41) 177
  warning message 177
Exception Threshold Field Details panel 61, 65
Exception Threshold Field Selection panel 60
EXCLUDE subcommand option 192
  filtering input 192
EXCPTDD data set 32
EXFILDD1 data set 32
explain 75
  authorization 78
  data set 33
  description 7
  distributed data 183
  streamlining processing 197
EXTRCDD1 data set 32

E
empty reports 11, 241
exception
  specifying 59
exception field
  changing the description 221
Exception File data set
  definition 58
  exception log 134
  definition 58
  description 5
Exception Log data set
  defining in JCL 58, 134
Exception Log File data set
  attributes 177
  defining in JCL 59, 134, 177
exception processing 175
  Accounting 134
  description 58
  output types 58, 176
  profiling 178
  statistics 171
  which thresholds to specify 66
exception profiling
  data sets involved 178
  purpose 178
  when to use 67
Exception Profiling panel 179
exception report 134
  Accounting 64
  definition 58
  exception threshold 134
  streamlining processing 197
Exception Threshold Category Selection panel 59
Exception Threshold data set
  attributes 178

G
Generation Data Group (GDG) 28
GLOBAL command
  description 34
  suppressing PRESORTED 194
  TIMEZONE option 209, 213
GROUP command
  defining OMEGAMON XE for DB2 PE identifier
  values 195
  description 34
  improving Save data set performance 233
  group-scope report 168, 186
    Accounting 190
    collecting input 190
    description 186
    example 188
    Statistics 189
  group-scope trace 186
  description 186
  group, data sharing 168

Index 257
headers
statistics 173
high water mark value 161
highlighting vii
historical data 197
streamlining processing 197

I
I/O activity
description 7
input 95
identification, thread 121
identifier
See OMEGAMON XE for DB2 PE identifier
IFCID (instrumentation facility component identifier)
description 25
frequency distribution log
description 5
IFCID 350
for SQL Activity trace 71, 75
IFCID 63
for SQL Activity trace 75
SQL statement text 75
IFCID option 23
IFCID Selection window 16
INCLUDE subcommand option 192
filtering IFCIDs 75
filtering input 192
information xi
input
I/O activity 95
SQL activity 69
INPUTDD data set 32
INSTALL parameter values 21
instrumentation facility
description 8
START TRACE command 21, 25
instrumentation facility component identifier (IFCID)
See IFCID (instrumentation facility component identifier)
Interactive Report Facility (IRF)
Accounting Report panel 42
Accounting Report Selections panel 42
Background DName Selections panel 43
composing a batch report command stream 39
example of generated job stream 44
example of producing an Accounting report 40
foreground and background processing 39
Interactive Report Selections panel 38, 40
Job Processing Selections panel 44
overview 37
Recall Selections panel 40
saving and recalling selections 40
selecting report sets 38
Session Options window 41
specifying default options 39
using BROWSE command 39
using COMPOSE command 40
Interactive Report Facility (IRF) (continued)
using EXCLUDE command 38
using GROUP command 38
using INCLUDE command 38
using LIST command 38
using OPTIONS command 39
using RECALL command 39
using RESET command 39
using SAVE command 39
interval (statistics) 160
INTERVAL option 197
streamlining processing 197
interval record 162
introduction to OMEGAMON XE for DB2 PE 1
IRF
see Interactive Report Facility (IRF) 37
IRF Save Selections data set
attributes 40
ISPF Online Monitor
access CRD panels 13
access to Collect Report Data panels 13
using 12
ISPF/PDF editor, using 35

J
JESMSGLG 240
Job Processing Selections panel 180
job summary log
description 5
JOBSUMDD data set 32
JSSRSDD (job summary VSAM data set) 32

K
keyboard shortcuts xii

L
labels vii
layout
adding a block 202
displaying 206
replacing fields 204
saving 207
selecting 200
tailoring Accounting and Statistics reports and traces 198, 208
LAYOUT subcommand option 195
library Web page xi
limiting input 191, 194
LIST command 34, 196
description 34, 196
LOCATION parameter 24
location, requester 122
locking
description 6
group-scope report 188
member-scope report 186
log, exception 134
logs
description 5

M
main functions of OMEGAMON XE for DB2 PE 1
main package
See MAINPACK identifier
MAINPACK Definition Member Editor panel 216
MAINPACK identifier
defining 215, 221
examples 217, 219
manage DB2 traces 8
member scope 120
member-scope report 166, 186
description 186
example 186
member-scope trace 186
description 186
menu
data set maintenance
defining exception thresholds 59
defining the MAINPACK identifier 215
specifying time zones 209
message
DPMLOG execution log 5
FPEA4531I 243
FPEA4532I 243
FPEA4534I 243
FPEC4015I 241
FPEC4020I 241
NO DATA TO REPORT 241
monitoring data sharing groups
etire group 188, 189
individual members 186, 187
multi-site report 122

N
Near-Term History Data Collector
using sequential data sets 30
non-DDF data
definition 121
nondistributed Accounting times 125
nondistributed transaction 121

O
OMEGAMON XE for DB2 PE 1
components 2
introduction 1
main functions 1
output types 5
OMEGAMON XE for DB2 PE identifier
See also AUTHID/PRIMAUTH identifier
See also MAINPACK identifier
See also ORIGAUTH identifier
AUTHID/PRIMAUTH
comparing original with primary 234, 235
DB2 START TRACE command 23
defining sets 195

OMEGAMON XE for DB2 PE identifier (continued)
filtering input using INCLUDE and EXCLUDE 192
MAINPACK 215, 221
ORIGAUTH 234, 235
OPTIMIZE subcommand option of REDUCE 236
ORDER processing
Accounting 135
Order subcommand option
in distributed environment 138
ORIGAUTH identifier 234, 235
comparing with primary authorization ID 234, 235
original authorization ID 234
See ORIGAUTH identifier
output data set for DB2 trace data
attributes 19
overview 1
overview of Performance Expert Agent for DB2 Connect
Monitoring (PE Agent) 3

P
pairs, record 164
panel
Accounting Report 42
Accounting Report Selections 42
Background DDname Selections 43
Collect Report Data 13
Exception Profiling 179
Exception Threshold Category Selection 59
Exception Threshold Field Details 61, 65
Exception Threshold Field Selection 60
Interactive Report Selections 38, 40
Job Processing Selections 44, 180
MAINPACK Definition Member Editor 216
Recall Selections 40
Time Zone Data Editor 210
User-Tailored Reporting Layout Generation 200
UTR Block Selection 201
UTR Field Selection 203, 204
UTR Layout Selection 201
UTR Save Layout Details 207
parallelism
effects on roll up 124
parallelism, query CP 123
parallelism, Sysplex query 123
parameter
INSTALL 21
LOCATION 24
performance data
generating, collecting, processing 8
using the workstation client 8
performance data collection 11, 25
DB2 START TRACE command 21, 25
INSTALL parameter values 21
using IFCIDs 25
using the workstation client 12
performance database 221
Accounting tables 223
audit tables 224
exception table 226
locking tables 227
performance database (continued)

Record Trace tables 227
Statistics table 229
structure 223
Performance Expert Agent for DB2 Connect Monitoring
(PE Agent) overview 3
performance warehouse 221
PLAN option 23
PRESORTED option suppressing 194
PRESORTED option, suppressing 194
primary authorization ID See PRIMAUTH/AUTHID identifier
PRIMAUTH/AUTHID identifier comparing original with primary 234, 235
DB2 START TRACE command 23
processing exceptions 175
program
Collect Report Data Batch 28
PTASKROL subsystem parameter 124
publications
 online xi
 ordering xi

Q
query CP parallelism 123
query parallelism 123
query parallelism, Sysplex 123

R
record pairs 164
record trace
description 7
record, delta 162
record, interval 162
records pair, DB2 Statistics 161
Redbooks xi
REDUCE subcommand 35
calculate ACWORK size 236
collection options 31
description 35
control performance 236
report
 accounting
 controlling the level of detail 195
 Accounting report example, using IRF 40
 composing JCL with IRF 39
 controlling the level of detail 195
 creating 31
description 5
empty 11
grope-scope 189, 190
how to generate 11
input 127
selecting report set 38
statistics
 controlling the level of detail 195
report set
 commands 34
description 6
generating
 using ISPF/PDF 35
 level of detail 8
 reporting process overview 8
 REPORT subcommand 35
description 35
 report, group-scope 168
 report, member-scope 166
 report, multi-site 122
 report, single-site 122
 report, user-tailored 198
 reports and traces
 input for Statistics 164
 requester location 122
 resource access
 tracking 106
 RESTORE subcommand 35
description 35
 RKD2PAR(COPT<ssid>)
collection options 31
 WRITEOPTION keyword 31
 RKO2DATA data set
 DGOETV41 member 65, 66
 RKO2DATA(DGOETV41)
different versions of 177
 sample Exception Threshold data set 177
 warning message 177
 roll up
 controlling 123, 124

S
sample 28
Save data set
 improving performance 233
Save Layout Details panel 207
SAVE subcommand 35
description 35
scope, member 120
screen readers and magnifiers xii
short report
 account
 generating with ISPF/PDF 35
tailoring 199, 208
single-site report 122
SMF (system management facility) 22
snapshot value 161
snapshot value (delta) 162
snapshot value (interval) 163
SQL activity 67
description 6
distributed data 183
input 69
SQL Activity trace
description 6
IFCID 63 75
SQL statement text 75
SQL statement 75
SQL statement text
  IFCID 350 75
  IFCID 63 75
SQL Activity trace 75
START TRACE
  collecting IFCID 350 75
START TRACE command 21, 25
  syntax 22
starting and stopping traces of a collect task 18
statistics
  description 6
  distributed data 183
  exception processing 171
  grope-scope report 189
  headers used 173
  how values are reported 175
  reporting values 175
  tailoring report and trace layouts 198, 208
  values reporting 175
Statistics 47
  functions 164
  input for reports and traces 164
  level of detail 165
  sample layouts 165
Statistics records pair 161
STEPLIB data set 31
subcommand 35
  See also auxiliary command
  See also command
description 35
FILE 35
REDUCE 35
  using CALCULATE option 236
  using OPTIMIZE option 236
REPORT 35
RESTORE 35
SAVE 35
TRACE 35
subsystem parameter
  ACCUMACC 123
  PTASKROL 124
support xii
Support Assistant xii
suspensions, class 3 127
syntax diagrams ix
SYasin data set 31
SYSOUT data set 32
Sysplex query parallelism 123
SYSPRINT message log
  See DPMLOG execution log
system management facility (SMF) 22
system overview 47
system parameters reports
data set 33
description 8

T
  tailoring (continued)
  Accounting and Statistics reports and traces 198, 208
  TDATA option 23
  thread 52
    selecting for DDF 184
  thread identification 121
  thread types 120
  thread, allied 121
  thread, allied-distributed 121
  thread, database access 121
  thread, DBAT-distributed 121
  threshold
    types 134
  threshold, exception 134
time zone
    editor 209
    example 211, 212
    specifying 209, 213
Time Zone Data Editor panel 210
times, distributed Accounting 125
times, nondistributed Accounting 125
TIMEZONE option 209, 213
  GLOBAL command 209, 213
TKO2SAMP(FPEZCRDJ) 28
TO subcommand option 191
  filtering input 191
TOP (ONLY) subcommand option 135
TOP list
  producing 143
top processing
  Accounting 143
trace
  See also START TRACE command
topology accounting
  controlling the level of detail 195
  controlling the level of detail 195
description 5
  input 127
  starting automatically 21
statistics
  controlling the level of detail 195
trace class
  description 25
trace classes 128
  30, 31, 32 75
trace classes, Accounting 124
Trace Configuration window 14
trace facility (DB2)
  See DB2 instrumentation facility
TRACE option 22
Trace Qualification window 17
Trace Status Summary window 20
TRACE subcommand 35
description 35
takes and reports
  input for Statistics 164
  traces, starting and stopping 18
transaction, distributed 122
transaction, nondistributed 121
translation
  See correlation translation
Trigger by Time window  18
troubleshooting  241
typeface vii

U
updates
  service xii
user identification
  See also AUTHID/PRMAUTH identifier
  See ORIGAUTH identifier
user-tailored report  198
User-Tailored Reporting (UTR)
  See UTR (User-Tailored Reporting)
User-Tailored Reporting Layout Generation panel  200
using the ISPF Online Monitor  12
using with Collect Report Data Batch program  28
utility activity
  description  7
UTR (User-Tailored Reporting)  198, 208
  Accounting report (short)  199
  block of data
    adding  202
    replacing fields  204
  description  198
  layout
    displaying  206
    saving  207
    selecting  200
UTR Block Selection panel  201
UTR field selection panel  204
UTR Field Selection panel  203
UTR Layout Selection panel  201
UTR Save Layout Details panel  207

V
value
  GROUP command  195
  INSTALL parameter  21
value, accumulated  161
value, Accumulated (delta)  162
value, Accumulated (interval)  163
value, high water mark  161
value, snapshot  161
value, snapshot (delta)  162
value, snapshot (interval)  163

W
window
  Data to Collect  16
DGOMAP10  20
DGOMAP30  14
DGOMAP31  16
DGOMAP32  16
DGOMAP33  17
DGOMAP40  18
IFCID Selection  16
Readers’ Comments — We’d Like to Hear from You

IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS
IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS
Reporting User’s Guide
Version 4.2.0

Publication No. SC19-2510-00

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.
Send your comments to the address on the reverse side of this form.
If you would like a response from IBM, please fill in the following information:

Name ___________________________ Address ___________________________

Company or Organization ___________________________

Phone No. ___________________________ E-mail address ___________________________