DB2 Performance Monitor for OS/390

Batch User's Guide

Version 6

SC26-9167-01

Note

Before using this information and the product it supports, be sure to read the information in "Appendix E. Notices" on page 279.

Second Edition, April 2000

This edition applies to Version 6 of IBM DATABASE 2 Performance Monitor for OS/390, a feature of IBM DATABASE 2 Universal Database Server for OS/390 Version 6 (5645-DB2), and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC26-9167-00.

Contents

About This Book

This book describes how to use IBM DATABASE 2(TM) Performance Monitor for OS/390(R) Version 6. DB2(R) PM is a performance analysis tool that helps you monitor and tune the following IBM DATABASE 2 products:

- IBM DATABASE 2 Universal Database Server for OS/390 (DB2 UDB for OS/390) Version 6, Program Product 5645-DB2
- IBM DATABASE 2 Server for OS/390 (DB2 for OS/390) Version 5, Program Product 5655-DB2
- IBM DATABASE 2 for MVS/ESA (DB2 MVS/ESA(TM)) Version 4, Program Product 5695-DB2.

DB2 PM Version 6 is a feature of DB2 UDB for OS/390.

Who Should Read This Book

This book is helpful to any user responsible for monitoring and tuning DB2. It provides you with an overview of the reporting and graphics capabilities of DB2 PM and gives you some guidelines for setting up your strategy for using the product.

After reading this book, you should be able to select the report sets most appropriate for your requirements. You should also be familiar with the two methods you can use to generate the commands and JCL necessary to create DB2 PM reports. The tuning information in this book helps you interpret the information on the reports. If you need more detailed information, refer to the *DB2 PM Report Reference*.

For information about using the DB2 PM Online Monitor to monitor an active DB2 system, refer to the *DB2 PM Online Monitor User's Guide*.

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 book or any other DB2 PM documentation, send your comments by using:

- Internet. The address is: swsdid@de.ibm.com.
- IBM Mail Exchange. The address is: DEIBM3P3 at IBMMAIL.
- The form at the back of this book. Return it by mail or fax, or give it to an IBM representative. The fax number is: +49-(0)7031-164892.

Be sure to include the name of the book, the version of DB2 PM, and, if applicable, the specific location of the text you are commenting on (for example, a page number or a table number).

How to Use This Book

This book is designed for both new and experienced users of DB2 PM. New users should read all parts of this book whereas experienced users can skip "Part 2. Overview of DB2 PM Batch" on page 7.

How This Book Is Organized

This book is divided into the following parts:

1. "Part 1. Migration from Previous Versions" on page 1
2. "Part 2. Overview of DB2 PM Batch" on page 7
3. "Part 3. How to Use DB2 PM Batch" on page 23
4. "Part 4. Monitoring and Tuning with DB2 PM Batch" on page 175
5. Appendixes.

Read "Part 2. Overview of DB2 PM Batch" on page 7 the first time you use DB2 PM.

"Part 3. How to Use DB2 PM Batch" on page 23 tells you how to use DB2 PM, streamline and customize DB2 PM processing, report by exceptions, and produce graphs.

"Part 4. Monitoring and Tuning with DB2 PM Batch" on page 175 shows how to interpret the information provided in the most commonly used DB2 PM reports.

Programming Interface Information

This publication is intended as a guide for using the DB2 PM Interactive Batch feature to monitor and tune DB2.

This publication also documents product-sensitive programming interface and associated guidance information provided by IBM DATABASE 2 (DB2) only.

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning DB2. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive programming interface and associated guidance information is identified where it occurs by the following marking:

Product-Sensitive Programming Interface

Product-sensitive programming interface and associated guidance information...

End of Product-Sensitive Programming Interface

Prerequisites

A working knowledge of, and experience with, either one of the following IBM DATABASE 2 products is required:

- DB2 Universal Database Server for OS/390 Version 6
- DB2 for OS/390 Version 5
- DB2 for MVS/ESA Version 4.

How to Read the Syntax Diagrams

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

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

  The ►►─── symbol indicates the beginning of a statement.

  The ───► symbol indicates that the statement syntax is continued on the next line.

  The ►─── symbol indicates that a statement is continued from the previous line.

  The ───►◄ symbol indicates the end of a statement.

  Diagrams of syntactical units other than complete statements start with the ►─── symbol and end with the ───► symbol.
- Required items appear on the horizontal line (the main path).

```
►►──required_item──────────────────────────────────────────────►◄
```

- Optional items normally appear below the main path.

```
►►──required_item──┬───────────────┬───────────────────────────►◄
                   └─optional_item─┘
```

  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.

```
                   ┌─optional_item─┐
►►──required_item──┴───────────────┴───────────────────────────►◄
```

- 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 main path.

```
►►──required_item──┬─required_choice1─┬────────────────────────►◄
                   └─required_choice2─┘
```

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

```
>>──required_item──┬──────────────────┬───────────────────────────><
                   ├─optional_choice1─┤
                   └─optional_choice2─┘
```

If one of the items is the default, it appears above the main path and the remaining choices are shown below.

```
                   ┌─default_choice──┐
>>──required_item──┼─────────────────┼────────────────────────────><
                   ├─optional_choice─┤
                   └─optional_choice─┘
```

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

```
                   ┌─────────────────┐
                   ▼                 │
>>──required_item────repeatable_item─┴────────────────────────────><
```

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

```
                   ┌─,───────────────┐
                   ▼                 │
>>──required_item────repeatable_item─┴────────────────────────────><
```

  If the repeat arrow contains a number in brackets, the number represents the maximum number of times that item can appear.

```
                   ┌─(5)─────────────┐
                   ▼                 │
>>──required_item────repeatable_item─┴────────────────────────────><
```

  A repeat arrow above a stack indicates that you can repeat the items in the stack.
- Keywords appear in uppercase (for example, FROM). Variables appear in all lowercase letters (for example, *column name*). They represent user-supplied names or values.
- If punctuation marks, parentheses, arithmetic operators, or other such symbols are shown, you must enter them as part of the syntax.

Summary of Changes

This section lists the changes to DB2 PM in Version 6. DB2 PM Version 6 supports all the functions available in DB2 PM Version 4 and Version 5. In addition, all new instrumentation data of DB2 UDB for OS/390 Version 6 is supported by the Workstation Online Monitor and batch reports.

- DB2 PM now shows time and events for the newly introduced triggers and user-defined functions.
- DB2 PM monitors the behavior of large objects and the new ROWID data type.

- Service task switch time and synchronous I/O time are broken down into various class-3 times.
- DDF threads are shown in more detail, such as information on the end user's user ID, the workstation name, and the transaction name, which you can use as ordering and filtering criteria to customize your report sets.

The following sections describe the components and functions that are new or enhanced for DB2 PM Version 6.

Workstation Online Monitor for Windows NT and OS/2

DB2 PM Version 5 introduced the Workstation Online Monitor for both the OS/2(R) and Windows NT environments. The Workstation Online Monitor offers some significant advantages:

- Eliminate the need to monitor through TSO
- Improve your efficiency with an easy-to-use graphical interface
- Monitor multiple DB2 subsystems concurrently
- Offer improved tuning recommendations on a per-field basis
- Let you submit DB2 commands from DB2 PM workstation windows
- Display historical data in enhanced System Health graphics.

For Version 6, the Workstation Online Monitor adds several new functions, such as:

- Thread qualification and thread sort
- Intervall processing
- Reporting of statistics delta
- Statistics data is provided for cached SQL statements so you can analyze and evaluate their efficiency.

DB2 PM Installer

In Version 6, you can install, migrate, and customize DB2 PM from your workstation using a graphical interface, which guides you through the installation sequence. You can activate the DB2 PM Installer as a subfunction from the DB2 Installer and

- Install DB2 PM and control the overall installation process
- Run SMP/E installation jobs.

You receive job status information dynamically, and you can edit JCL, perform job cleanup, and examine job output from the workstation. The DB2 PM Installer enhances your productivity significantly whether you are installing DB2 PM for the first time or are an experienced installer.

Part 1. Migration from Previous Versions

This part describes the differences and requirements when migrating from DB2 PM Version 5 and Version 4 to Version 6.

It is divided into the following chapters:

- Chapter 1. Migrating from DB2 PM Version 5 describes the changes for users migrating from Version 5 to Version 6.
- Chapter 2. Migrating from DB2 PM Version 4 describes the changes for users migrating from Version 4 to Version 6.

Chapter 1. Migrating from DB2 PM Version 5

This chapter describes major differences and requirements when migrating from DB2 PM Version 5 to DB2 PM Version 6.

Migrating Save Data Sets

New releases of DB2 bring more accounting and statistics timers and counters. DB2 PM supports them all in save data sets. If you want to use existing accounting and statistics save data sets, you must migrate them to Version 6 format using the save-file utility.

Refer to the specific chapters on save data sets migration in the *DB2 PM Report Reference* for a detailed description on how to migrate accounting and statistics save data sets.

Note: If you report newly created data together with save data that was migrated and restored, you get two separate accounting reports for the same location or subsystem.

File Data Sets

DB2 PM supports all the new performance data in its file data sets. This is why the field offsets of the file data sets have changed. The new samples for the CREATE, LOAD, and QUERY statements cannot be used with the Version 5 file data sets. Neither can the existing CREATE, LOAD, and QUERY statements be used with the new file data sets.

For detailed information on how to migrate the performance database, refer to the *DB2 PM Report Reference*.

Performance Database

Only DB2 PM Version 6 related data or data that has been migrated to Version 6 can be loaded into the Version 6 performance database. For detailed information on the performance database, refer to the *DB2 PM Report Reference*.

Chapter 2. Migrating from DB2 PM Version 4

This chapter describes major differences and requirements when migrating from DB2 PM Version 4 to DB2 PM Version 6.

Migrating Save Data Sets

New releases of DB2 bring more accounting and statistics timers and counters. DB2 PM supports them all in save data sets. If you want to use existing accounting and statistics save data sets, you must migrate them to Version 6 format using the save-file utility.

Refer to the specific chapters on save data sets migration in the *DB2 PM Report Reference* for a detailed description on how to migrate accounting and statistics save data sets.

Note: If you report newly created data together with save data that was migrated and restored, you get two separate accounting reports for the same location or subsystem.

File Data Sets

DB2 PM supports all the new performance data in its file data sets. This is why the field offsets of the file data sets have changed. The new samples for the CREATE, LOAD, and QUERY statements cannot be used with the Version 4 file data sets. Neither can the existing CREATE, LOAD, and QUERY statements be used with the new file data sets.

For detailed information on how to migrate the performance database, refer to the *DB2 PM Report Reference*.

Performance Database

Only DB2 PM Version 6 related data or data that has been migrated to Version 6 can be loaded into the Version 6 performance database. For detailed information on the performance database, refer to the *DB2 PM Report Reference*.

Report Command Language

Utility Trace

The LEVEL option of Utility Trace has been dropped and a new WORKLOAD option has been added. As with SQL Activity, workload blocks can now be switched on or off individually. The detailed PHASE information and the detailed BIND information is now contained in a new workload block. For Utility Trace commands, change any JCL containing the LEVEL option as follows: UTILITY TRACE LEVEL(LONG) translates to UTILITY TRACE WORKLOAD(ALL), and UTILITY TRACE LEVEL(SHORT) becomes UTILITY TRACE WORKLOAD(NONE). WORKLOAD(NONE) can be omitted.

Utility Report

The LEVEL option of Utility Report has been dropped. The new Utility Report does not show any workload information. For Utility Report commands, remove any LEVEL options from your JCL.

Part 2. Overview of DB2 PM Batch

Chapter 3. Introduction to DB2 PM Batch

IBM DATABASE 2 Performance Monitor for OS/390 Batch (DB2 PM Batch) is a performance analysis tool for DB2. It presents information about DB2 performance in reports, graphs, and data sets. DB2 PM Version 6 is a feature of DB2 UDB for OS/390 Version 6.

You can use DB2 PM's output to determine the overall performance of a DB2 subsystem or the performance of an application (an individual program or a set of programs).

The system-wide performance data shows information about, amongst other things, CPU times, buffer pool usage, locking, log and I/O activity. For a particular application, you can find out, for example, the elapsed time, the time spent in DB2, the time it was suspended, the read and write activity involved, the locks obtained, and the SQL statements executed.

By analyzing DB2 PM output, you can identify areas where tuning is required to enhance the performance of DB2.

Purpose and Function

DB2 generates data about its own performance, called instrumentation data, but it has no reporting facility that you can use to analyze this data. DB2 PM provides you with the capability to gather, analyze, and report on DB2 instrumentation data.

The primary objective of DB2 PM is to report DB2 performance information in a form that is easy to understand and analyze.

You can use DB2 PM to:
- Determine DB2 subsystem performance and efficiency
- Identify and resolve potential problems
- Tune DB2
- Measure an application's performance and resource cost
- Measure an application's effect on other applications and on the system.

DB2 PM provides the performance information in various levels of detail. As a rule, you should first request reports showing the overall performance of a system or an application. If any performance problems are evident, you can produce a more detailed report for that specific area. If this information is not sufficient to solve the problem, you can produce even more detailed reports.

DB2 PM Batch gives a “historical” view of DB2 performance; if you want to monitor the performance of a DB2 subsystem while it is active, use the DB2 PM Online Monitor.

How the Performance Data Is Generated, Collected, and Processed

The DB2 trace facility, also called the DB2 instrumentation facility, gathers information about data and events in the system. After DB2 has collected and externalized this data, DB2 PM reads it and generates reports, graphs, and data sets from it.

Start the DB2 trace facility by issuing a START TRACE command. When you issue the command, DB2 starts to collect performance records in a data set you have specified as the trace destination.

DB2 Trace Data

When you invoke the DB2 trace facility you can specify the kind of performance information you want to collect. The data consists of different performance data record types, called instrumentation facility component identifiers (IFCIDs). An IFCID represents a significant DB2 event. The IFCIDs are grouped into classes and the classes are grouped into types.

In the START TRACE command, first specify the type of data, then the classes within that type. You can also specify particular IFCIDs.

Type is the broadest categorization of performance data. The types used by DB2 PM Batch are:

- Statistics
- Accounting
- Audit
- Performance
- Global.

These different types of trace data are divided into classes. Class defines a certain group of events or data within the type.For example, accounting trace class 1 records elapsed and CPU times spent in the application, class 2 records elapsed and CPU times spent in DB2, class 3 records suspension times, and class 5 records Instrument Facility Interface (IFI) related times.

A class consists of one or more IFCIDs. For example, the IFCIDs belonging to performance class 6 show locking information. Some of them are IFCIDs 20, 44, and 45. IFCID 20 shows page locking information, IFCID 44 lock suspension information, and IFCID 45 shows lock resumes.

One IFCID can belong to more than one class or type. There are also some IFCIDs that do not belong to any class or type.

What Trace Data to Collect

Collecting instrumentation data is a performance overhead, so you should carefully consider which traces to run.

You are recommended to regularly collect accounting class 1 and 3 and statistics class 1, 3, and 4 data.

You should also consider collecting accounting class 2 data. Class 2 data provides important information about DB2 times.

If you want to report on packages and database request modules (DBRMs), you need to start accounting class 7 or 8.

Accounting class 5 is needed if you want to report on IFI or data capture activity.

Start particular traces and classes only when there is a specific reason for it.

Refer to “DB2 Instrumentation Data” on page 36 for more information about the trace classes needed for different DB2 PM reports.

How DB2 PM Processes the Data

When you request a report, DB2 PM reads data from the data set you specified as the input file and converts it into a format suitable for report generation. It then generates the reports according to the DB2 PM commands you specified.

You can keep the processed input data in the DB2 PM output data set DPMOUT if you wish to do so. The DPMOUT data set can be used as input to DB2 PM. For more information, refer to “Do You Need a DPMOUT Data Set?” on page 112.

In producing explain information, DB2 PM connects to the active DB2 subsystem, reads the catalog and PLAN_TABLE, and gathers the information into reports.

How DB2 PM Reports the Data

DB2 PM generates reports, data sets, graphs, and logs. The output generation is controlled by means of DB2 PM commands.

Report Sets

DB2 PM output is grouped into report sets, each associated with a particular type of data. The report sets consist of reports, traces, and data sets.

Each report set shows performance data in different levels of detail and for different areas of performance. The report sets are:
- System parameters
- Statistics
- Accounting
- Explain
- SQL activity
- Utility activity
- Locking
- I/O activity
- Record trace
- Audit.

The *system parameters* report showsinformation 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. The report is indispensable in tuning your subsystem because the configuration of the system directly affects its performance.

Statistics summarizes system-wide performance data,whereas *accounting* summarizes information for particular applications. Both report sets show information for various areas of performance, such as elapsed times and high-level information about SQL activity, locking, and buffer pool utilization.

Statistics and accounting reports are used to determine the efficiency of the subsystem or application, and they often provide enough information for you to resolve performance problems. If they do not, they often indicate the area where the problem can be found. You can then generate more detailed reports.

Accounting and statistics data can be reported using exception processing. Exception processing provides a means of highlighting fields containing values that fall outside user-specified limits. This helps you focus on potential problems in the subsystem or in a thread.

The *explain* reports provide information aboutthe access path selected by DB2 for a particular SQL statement. This information is useful in application design and problem determination.

The next level of detail is provided by the *SQL activity* and *utility activity* reports. If the problem seems to be with the SQL statements, you can run SQL activity reports. The SQL activity report set identifies the SQL statements and the associated workload. Or, if a DB2 utility or bind seems to be the cause of the problem, you can request utility activity reports.

Locking and *I/O activity* reports represent the fourth level of detail. So, if the other reports indicate that the problem is with locking activity, you should generate the appropriate locking report. Use the I/O activity reports to investigate problems associated with read, write, and logging activity.

If none of the above reports can identify the cause of the problem, and this is rarely the case, a *record trace* can be run. The record trace formats selected instrumentation records into readable report entries.

Audit reports are not, strictly speaking, performance reports; they show information about authorization and the users of specific resources.

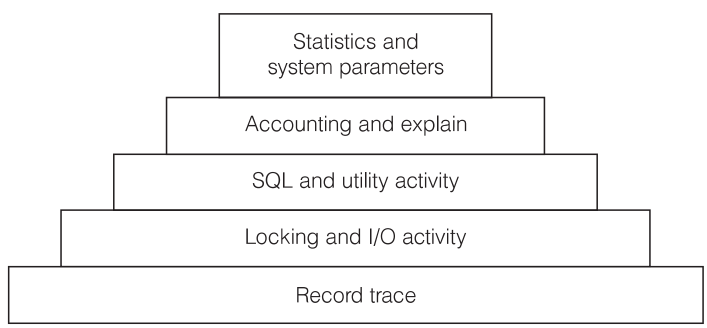

Figure 1. Report Sets in Order of Detail

Output Types

DB2 PM Batch generates traces and reports, as well as data sets, graphs, and logs.

Traces and Reports

Traces show individual DB2 events, for example, for a particular thread. Depending on the report set 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.

Reports show these events summarized by DB2 PM identifiers, such as primary authorization ID or plan name. You can, for example, produce an accounting report that shows all threads summarized for every individual plan.

Data Sets

Formatted data can be stored in data sets suitable for loading into DB2 tables. The data in DB2 tables can be put to different uses; for example, it can be used to produce tailored reports using a reporting facility such as the IBM Query Management Facility (QMF(TM)).

You can also load reports and traces into DB2 PM's performance database for additional analysis. For further information on this performance database, see the *DB2 PM Report Reference*.

Graphs

Graphs are generated and viewed online using the Interactive Report Facility (IRF). You can also calculate frequency distributions for selected DB2 PM fields and view the results using the graphics function. For more information, refer to "Chapter 13. Producing Graphs" on page 157.

Logs

Every time you run DB2 PM, it generates logs showing information about its processing, provided you have specified appropriate ddnames for them (refer to "DDNAMEs for DB2 PM Data Sets" on page 49 for more information). The DB2 PM logs are:

- *The Exception Log* identifies accounting and statistics records with at least one field outside user-specified limits.
- *The IFCID Frequency Distribution Log* provides counts of input records by IFCID.
- *The Job Summary Log* provides a summary of events during DB2 PM execution.
- *The DPMLOG Execution Log* shows messages issued during DB2 PM processing.

Commands

DB2 PM report generation is controlled by means of commands. You can either use the menu-driven DB2 PM interface, the Interactive Reporting Facility, to produce the reports, or you can enter the required DB2 PM commands and the JCL using a conventional editor.

The command you would usually use first, before you request any reports, is GLOBAL. It filters the input data and sets defaults for other commands. You can specify, for example, the start and end times for the data to be reported and records to be included and excluded. You should always carefully consider what data you need to process in order to avoid unnecessary performance overhead.

You request the reports by specifying the report set command and subcommands.

The report set command is the name of the report set or its abbreviation.

After you have specified the report set, you indicate how you want the data for that report set presented:

- If you want a summarized listing of the data, use the REDUCE and REPORT subcommands. REDUCE consolidates DB2 events with the same DB2 PM identifiers into one record. The reduced data is used as input to REPORT, which produces a summarized listing of the data.
- If you want to save reduced accounting and statistics data in a data set, use the SAVE subcommand. The save-file utility converts the data set to a format that is suitable for loading into DB2.
- If you want to report previously saved data, use the RESTORE subcommand.
- If you want to produce reports that list DB2 events individually, use the TRACE subcommand.
- If you want to store data about individual DB2 events in data sets or DB2 tables, use the FILE subcommand.

- If you want to produce frequency distribution graphs, use the DISTRIBUTE command.

The subcommands have various options; you can specify, for example, the level of detail on the reports or traces, how you want the data to be ordered, and which records are to be included and excluded.

The system parameters report set is different from the other report sets in that it is generated automatically, provided you have specified an appropriate DD statement for it.

For more information about commands, refer to “Chapter 6. Running DB2 PM Jobs” on page 47 and “Appendix D. How to Use DB2 PM Command Language” on page 277.

Overview of DB2 PM Reporting Process

Figure 2 summarizes the main elements and functions involved in DB2 PM processing. The top part of the figure shows the various inputs to DB2 PM 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.

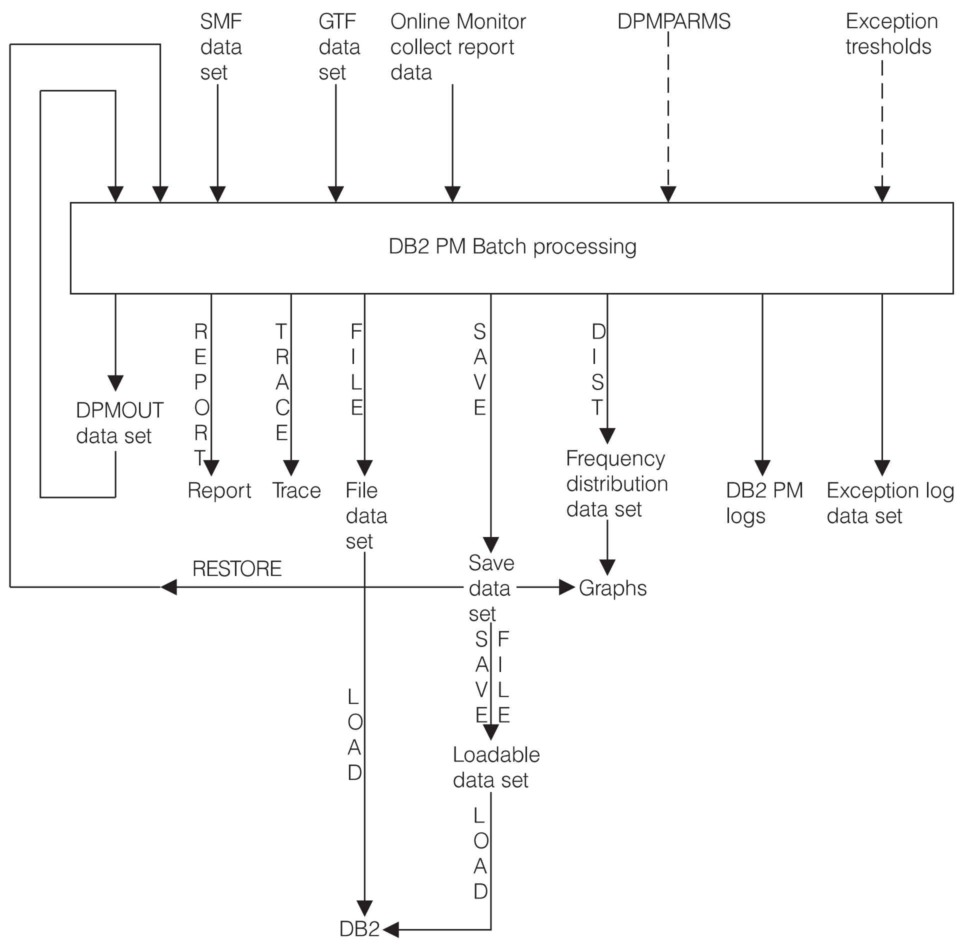

Figure 2. DB2 PM Reporting

For information about input data sets, read “Chapter 5. Collecting Performance Data” on page 27, and for information about DPMPARMS, read “Chapter 10. Customizing DB2 PM Functions” on page 113. Exception thresholds are described in “Chapter 8. Exception Reporting” on page 95.

You can find more information about commands in “Chapter 6. Running DB2 PM Jobs” on page 47. Some of the output types are described in “Chapter 7. Generating Commonly Used Reports” on page 61. Graphs are described in “Chapter 13. Producing Graphs” on page 157.

Chapter 4. Monitoring and Problem Determination

This chapter describes how to monitor a DB2 subsystem and how to determine performance problems using both the Batch reports and the Online Monitor functions of DB2 PM.

You can use DB2 PM for the continuous or periodic monitoring of a DB2 subsystem as well as for determining specific performance problems in DB2. DB2 PM can help you in:

- Determining how an application will perform or is performing over a period of time
- Indicating where there are tuning opportunities in your system
- Distinguishing between subsystem and application problems
- Monitoring an application in a detailed manner so you can identify problem areas
- Analyzing constraints acting on an application
- Determining the performance impact of any adjustments made within the DB2 subsystem
- Determining whether performance objectives are being met.

DB2 PM offers different ways for you to monitor your subsystem depending on whether you want to see current or past activity: the Online Monitor is used to monitor an active subsystem as well as to view events that happened in the recent past, whereas the Batch reports are used to examine performance problems in the more distant past and trends over a period of time.

The amount of data generated for monitoring a DB2 subsystem is vast, so limiting the amount of data to show only potential problem areas is essential. DB2 PM offers various ways of reducing the amount of data that needs to be examined, the most important of which is exception processing. Exception processing makes it easy for you to focus on possible performance problems by highlighting data that is outside limits you have specified. It is available in both Batch reporting and the Online Monitor.

The recommended approach is to monitor an active DB2 subsystem using Online Monitor exception processing (you do not need to be logged on to have exception processing running) and to regularly generate statistics and accounting exception reports. Online Monitor exception processing alerts you to performance problems as soon as they occur, and the accounting and statistics reports give you a detailed picture of application and system performance over a period of time.

Deadlock and timeout participant details are available online through event exception processing. You should also consider generating deadlock and timeout traces regularly, because in this way information is available to help you investigate any locking problems in detail. There is no significant performance overhead on the DB2 side in collecting the data for these reports.

The best way for you to investigate performance trends is by using IRF graphics and by producing accounting and statistics reports ordered by interval.

To detect problems as they occur, use the Online Monitor periodic exception processing. When you detect poor thread performance, you can use the comprehensive performance data shown on thread and statistics panels to identify and resolve the problem. If you think the problem is caused by SQL, you can analyze the access path using the online explain function.

If the problem occurred in the recent past, you can use the Online Monitor HISTORY command to view the events surrounding the problem without having to recreate it.

If the panels do not provide enough information to solve the problem, you can use the Online Monitor to collect instrumentation data for batch reports. You can specify the data collection to be triggered by exception thresholds; in this way you can minimize the time high-volume and high-cost traces are active and ensure that the data needed is collected at the right time.

DB2 PM provides a comprehensive set of reports with different levels of detail and for different areas of performance. This represents a top-down approach to problem determination: the most generic reports indicate the problem area and, if necessary, more detail can be shown to narrow down the cause of the problem.

The DB2 Operating Environment

The performance of a DB2 transaction or query is dependent not only on the performance of the DB2 subsystem, but also on the performance of the transaction manager, such as CICS(R) or IMS, and the MVS(TM) system itself. Therefore the environment in which the DB2 subsystem is operating should be tuned before DB2 is tuned.

For example, if the MVS system is overloaded, tuning a DB2 subsystem is unlikely to improve DB2 system performance. DB2 performance can only be improved by reducing or balancing the load of the MVS system.

Specialized tools are available to monitor the different system components:
- CICSplex System Manager for CICS
- IMS/VS DC Monitor or IMSPARS for IMS
- RMF(TM) for MVS.

The relationship between the different systems and performance tools is complex and it is not within the scope of this book to describe them. Refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for more information about them. For the purposes of this book it is assumed that the environment in which the DB2 subsystem is operating is well tuned.

Performance Objectives and Exception Processing

Before you can start monitoring the system, you should define your performance objectives on the basis of the business needs, the workload for the system, and the resources available. Typically, the objectives would include acceptable response times, average throughput, and system availability.

These objectives are usually formalized in service-level agreements between the users and the data processing groups in an organization. The agreements can include expectations of, for example, query response times and transaction throughput.

With DB2 PM, you can monitor how well these objectives are being met.

The most efficient way to do this is to set limits, exception thresholds, for key fields that reflect your performance objectives using exception processing.

Exception profiling can assist you in establishing exception thresholds. This facility sets exception thresholds automatically based on your application configuration. For reports, the accounting TOP option is also useful in determining accounting exception thresholds.

You can, for example, monitor response times by setting exception thresholds for class 1 and class 2 elapsed times to reflect the acceptable response times for your environment. Class 1 elapsed time shows the thread time (from thread creation to thread termination) and class 2 time shows the time DB2 spent processing SQL statements.

Monitoring

The key to effective performance monitoring is in identifying unusual situations and thereby limiting the amount of data that needs to be examined. In addition to exception processing, DB2 PM offers various other ways of filtering the data and highlighting potential problems.

The following are available in Batch reporting:
- Filtering data by date and time (FROM/TO)
- Filtering data by identifiers such as user ID (INCLUDE/EXCLUDE)
- Filtering data by resource usage (TOP)
- Ordering data on reports by interval (INTERVAL)
- Summarizing and sorting data on SQL reports
- Tailoring report layouts for your own needs (UTR).

The following are available in the Online Monitor:
- Qualify and sort functions
- History data collection with qualifications.

Monitoring Using Reports

You can filter the data shown on reports by using the INCLUDE/EXCLUDE and FROM/TO filters to show, for example, information only for certain plans, authorization IDs, or locations within specified times.

Another way to limit the data that needs to be examined is to use the *accounting TOP option* to obtain a high water mark type of reporting on resource usage. The TOP lists, printed at the end of an accounting report or trace, identify, for example, the threads or users that have required the most use of the resources specified in the TOP option. Alternatively, you can use the *TOP ONLY option* to filter entries based on resource usage and produce a report showing only entries with the highest resource usage.

By ordering your accounting and statistics reports by *interval* you can summarize data for certain periods. It can be useful, for example, to summarize data for the peak periods during the day.

When you produce SQL activity reports and traces, potential problems can be highlighted by *sorting* and *summarizing* the information within the report and trace entries by various criteria. For example, a problem cursor can be identified by summarizing SQL activity by cursor and ordering the cursors by TCB times. The sorted entries provide an easy way to identify SQL statements that might be causing performance problems.

You can tailor your own report and trace layouts using the user-tailored reporting feature (UTR). This function gives you full control over the volume, contents, and layout of your traces and reports.

Monitoring Using the Online Monitor

The Online Monitor qualify and sort functions can help you limit and prioritize the threads listed on the Thread Summary panel.

Use the *qualify function* to filter threads by DB2 PM identifiers, thread status, and thread type, and to effectively reduce the amount of data that needs to be examined. For example, if you want to view the active threads in lock wait status only, you can qualify the threads by selecting the *In lock wait* field on the DB2 Thread Qualification Parameters window.

Use the *sort function* to specify the order in which the threads are listed. For example, if you want to view the threads that are spending the most time within DB2, you can sort the threads by class 2 time in descending order. Threads in exception status are automatically sorted to the top of the list.

Observing Trends

You can use DB2 PM reports and graphs to summarize data over periods of several days, weeks, or months to observe trends in performance.

In monitoring trends, you should pay special attention to peak periods of activity, both for new applications and for the system as a whole. During peak periods, constraints and response-time problems are most evident.

Some trends to look for are:

- Increases in response times, number of I/Os, resource contention, and CPU usage
- Changing workload patterns over a period
- Changes in the transaction distribution and frequency
- Changes in the SQL activity pattern.

Determining Problems

When you find that there are performance problems when you are monitoring the system, you can use various panels and reports to investigate the cause of the problems.

Problems Detected in Periodic Exception Processing

If you are using periodic exception processing in the Online Monitor and you are notified about a problem, the best way to find out what caused it is to examine the thread activity panels, especially thread diagnosis, or statistics panels, depending on the type of problem (see Figure 3 on page 21).

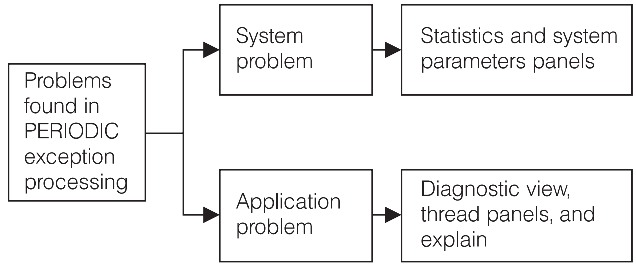

Figure 3. Problems Found in Periodic Exception Processing

To view the past events surrounding the problem online, you can use the HISTORY command. The amount of historical data available is determined by installation-defined options. Note, however, that the batch reports are more comprehensive than the Online Monitor panels. Therefore, if the panels do not provide enough information to solve a problem, the Online Monitor collect report data function should be used to gather information in a data set for input to the batch reports.

Problems Detected in Exception Event Processing

If you detect a problem using the online event exception processing and the Online Monitor panels do not provide sufficient information to determine the cause, then you should collect the appropriate trace data to produce locking reports, I/O activity reports, audit reports, or record traces. The following table shows you how to proceed when you analyze a specific event exception:

Exception	How to proceed
Deadlock or **Timeout**	If deadlocks or timeouts occur too often, generate a Lockout report to see which applications and objects are affected. Having identified the objects and applications causing the deadlocks or timeouts, use EXPLAIN to understand the locking behavior of the SQL statements or consider reorganizing the database.
EDM Pool Full	First check online or batch Statistics to get more information about the EDM Pool situation. More details are provided in the I/O Activity EDM Pool report.
Authorization Failure	If authorization failures occur too often, generate an Audit authorization failure report for details.
Thread Commit Indoubt	Run Record Trace on the Statistics Class 4 IFCIDs to see details of communication problems. These are likely to be either VTAM(R) or DB2 internal problems.
Coupling Facility Rebuild	Use the COLLECT command to automatically start tracing IFCID 268 (CF rebuild end) when a coupling facility rebuild starts, and run Record Trace for this IFCID. You can get more details about coupling facility behavior from RMF reports.

Problems Detected in Exception Reports

If you use the accounting and statistics exception reports to monitor your system, you can often detect the cause of a performance problem using the comprehensive information they offer without producing other reports. You should produce these reports using the TOP option in accounting or the INTERVAL option in both accounting and statistics so that you can immediately focus on potential problem areas. Sometimes, however, you require more detailed reports to determine the exact cause of a problem.

System Problems

If exception processing indicates problems in system-wide resource usage and a statistics trace does not clarify the reason for the problem, but points to EDM pool or logging activity, consider running I/O activity reports. Or, if the statistics trace indicates a problem with binds, you should generate utility activity reports. If the number of deadlocks is high, you should run locking reports. If none of these report sets offer adequate information to determine the cause of the problem, you can run a record trace to format the individual instrumentation records.

Application Problems

If exception processing indicates an application-related problem (usually an elapsed time problem for an application or a user), use explain reports to determine the access path of the suspected plan and, if necessary, generate accounting traces with the TOP option. If accounting and explain have not identified the reason for poor SQL performance, use SQL activity for detailed information on the specific statement such as scans or I/O per page set or sort specifics. If the accounting trace indicates a locking problem, run locking reports, or, if the problem seems to be with binds or DB2 utilities, run utility activity reports. As with system-related problem determination, you can run a record trace if none of these report sets provide adequate information to determine the cause of the problem.

Part 3. How to Use DB2 PM Batch

Chapter 5. Collecting Performance Data

DB2's instrumentation facility component (IFC) provides a trace facility that is used to record DB2 data and events. The collected instrumentation data is used as input to DB2 PM.

If you want to monitor a specific problem or the performance of a particular application, use the Online Monitor to collect the input data for your reports. All you need to do is specify the report set for which you want to collect data and the data set where it is to be collected. You can then use this data set as input to reports. You can start the Online Monitor report data collection manually, or you can set it to start automatically at a specified point in time or when an exception threshold is reached or an exception event is encountered.

For regular monitoring, you can either set the trace facility to start automatically when DB2 is started by using the INSTALL parameters of the DB2 Tracing panel (DSNTIPN), or you can invoke the trace facility by issuing a DB2 START TRACE command.

Usually the data is collected to the SMF or GTF buffer and then written to an SMF or GTF data set. Both SMF and GTF are service programs that provide a means of recording performance data. SMF is usually used for continuous monitoring; and GTF is used for monitoring a specific problem when the volume of the data is large.

Collecting Data Using the Online Monitor

You can use the DB2 PM Online Monitor to write performance data to a data set to be used as input to 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.

In order to collect performance data, you first configure a collect task. In each collect task, you specify the type of data 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 a data set you have specified.

You can configure and start up to four independent collect tasks. With each task you can collect trace data for:
- One or more DB2 PM 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.

Collecting Data

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.

After a collect task has been configured, you have to start it to begin the triggering of DB2 traces.

You can stop DB2 traces immediately by stopping the collect task.

Accessing the Collect Report Data Panels

You can access the collect report data panels from within the Online Monitor by selecting option 6 (*Collect Report Data*) from the DB2 PM Online Monitor Main Menu, by typing `COLLECT` on the command line, or by pressing **F17** (Collect).

The Collect Report Data panel (shown in Figure 4) is the first panel you see.

```
DGOMAP00                  Collect Report Data

PMO1DLOC           DSN1 V6

For any trace task enter one of the following actions:

  1=Configure
  2=Start
  3=Display
  4=Stop

  Task Description                              Status
_ Collect data for accct/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 4. Collect Report Data Panel

Using this panel you can:
- Display a window to configure the collect tasks (1) to collect data for DB2 PM reports and limit the collection of DB2 trace data to time periods or events of interest.
- Start a collect task (2) after it has been configured. Starting a collect task enables the start and stop criteria that were previously specified.
- Display a window to view the status of a collect task and any messages issued by that collect task (3).
- Stop a collect task and associated DB2 traces (4). Stopping a collect task prevents the triggering of DB2 traces and stops all traces that were previously started by the task.

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

In case you get a task error in the *Status* field, use the DB2 DISPLAY TRACE command to check if the traces are still running. If necessary, you can use the DB2 STOP TRACE command to stop the trace. Before you restart the appropriate collect report data function, you must exit the Collect Report Data panel (by pressing **F3**) and, in a second step, exit the DB2 PM Online Monitor Main Menu (by pressing **F3** again) to return to the IBM Database 2 Performance Monitor menu. This way you stop the four asynchronous tasks that were automatically set up when you first selected option 3 (*View online DB2 activity*) from the IBM Database 2 Performance Monitor menu.

Configuring a Collect Task

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

To configure a collect task, type 1 (Configure) next to a collect task on the Collect Report Data panel and press **Enter**. The Trace Configuration window is displayed on Figure 5. Use this window to specify the trace trigger method and the types of data to be collected.

```
DGOMAP30                 Trace Configuration

Task description . . . . . . : Collect data for acct/stats/audit
                                                          More:  - +
Trigger by . . . . . . . . . . 1   1=Time
                                   2=Periodic exception
                                   3=Exception event
                                   4=Immediate start

Enter one or more selection characters to start DB2 traces for specific
DB2 PM report sets or overtype with a blank to delete the selection.

>  Accounting
/  Audit
_  I/O Activity
_  Locking
_  Record Trace
_  SQL Activity
/  Statistics
_  System Parameters
_  Utility Activity


Enter one or more selection characters to qualify the data collection
or overtype with a blank to delete the selection.

/  Data Type
/  IFCID
/  Requesting Location, Plan name and Authid

 512  OP Buffer size (K-bytes)

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

Figure 5. Trace Configuration Window

Collecting Data

Before you can start a collect task, you must 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 DB2 PM 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 DB2 PM 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, IFCIDs, or DB2 PM identifiers. 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 through 1 024.

Selecting the Report

If you selected the *Data Type* field on the Trace Configuration window, the Data to Collect window (Figure 6) is displayed once for each selected report set that has more than one data type. Use this window to select the data types to be collected by the collect task for a specific report set.

The Data to Collect window shown in Figure 6 uses an example for audit.

```
DGOMAP31                      Data to Collect            Row 1 to 7 of 7

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 6. Data to Collect Window

If you select the *Select/Deselect all* field, all data type fields listed on this window are selected. If you type a blank in the *Select/Deselect all* field when all data type fields are selected, all fields on this window are blanked out.

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

Selecting the IFCID

If you selected the *IFCID* field on the Trace Configuration window, the IFCID Selection window is displayed (Figure 7). This window lets you exclude certain IFCIDs that would normally be collected for the selected report sets and data type. Note that this window is not shown if only one IFCID was collected for the previous selections.

```
DGOMAP32                     IFCID Selection          Row 1 to 8 of 12

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

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

_   Select/Deselect all

    IFCID  Description
_     1    System Services Statistics
_     2    Database Statistics
_     3    Accounting data
/    55    Set current SQLID
_    83    End of identify
_    87    End of signon
_   105    DBID/OBID for database and tablespace translation
_   107    Data set open/close information

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

Figure 7. IFCID Selection Window

As in the previous panel, you can use the *Select/Deselect all* field to select or deselect all fields on this panel.

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

Filtering the Data

If you selected the *Requesting Location, Plan name and Authid* field on the Trace Configuration window, the Trace Qualification window is displayed (Figure 8 on page 33).

Use this window to filter the data to be collected by the collect task. Only the data from threads matching the trace qualification criteria entered on this window is collected. These trace qualification criteria are also used if thread termination was indicated in the Trigger by Time window (Figure 9 on page 34).

```
DGOMAP33                    Trace Qualification

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


Change values as desired:

Requesting Location . . . . . . _________________  _________________
                                _________________  _________________
                                _________________  _________________
                                _________________  _________________

Plan name . . . . . . . . . . . ________  ________  ________  ________
                                ________  ________  ________  ________

Authid  . . . . . . . . . . . . ________  ________  ________  ________
                                ________  ________  ________  ________

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

Figure 8. Trace Qualification Window

Specify the name of the requesting location, plan name, and authorization ID, then press **Enter**.

Note: 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. For details, refer to *IBM DB2 Universal Database Server for OS/390 Version 6 Command Reference*.

Starting and Stopping Traces

You can trigger DB2 traces to start by time, periodic exception, exception event, or immediately. After you have specified the trace criteria, a window is displayed that lets you complete the start criteria and define the stop criteria. Which window is displayed depends on what you specified in the *Trigger by* field on the Trace Configuration window.

These windows are used to specify the criteria that must be met before the collect task is automatically started or stopped. They 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 may be lost when a buffer overrun occurs.

The Trigger by Time window (Figure 9 on page 34) is displayed if you have 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. With this window you can set the DB2 traces to start at a specified time, and to stop after a specified number of minutes have passed, number of records or IFCIDs have been collected, or a thread matching the trace qualification criteria has terminated.

```
DGOMAP40                    Trigger by Time

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

Output Data Set for DB2 trace data to be written to
  Name  . . . . . . . . . . . . DATASET1_______________________________
  Disposition . . . . . . . . . 1   1=Append
                                    2=Overwrite
                                    3=New

Start the DB2 traces at the following time
0   :  0   :  0   (hh:mm:ss)

Stop the DB2 traces when any of the following conditions occur
_  Elapsed time  . . . . . . . . . . . . . . . . 0      (seconds)
_  Number of records collected . . . . . . . . 0

Additional stop conditions
_  Thread termination
_  Number of IFCIDs collected  . . . . . . . . 0
   For IFCID . . . . . . . . . . . . . . . . . 55   +
       Event: Set current SQLID

Command ===> __________________________________________________________
 F1=Help    F2=Split   F3=Exit    F4=Prompt  F7=Up      F8=Down
 F9=Swap   F12=Cancel F16=Look
```

Figure 9. Trigger by Time Window

All trace data collected by the collect task is written to the data set specified on this window.

If you specify a disposition of 3 (new), the data set is dynamically allocated with the following attributes:

RECFM:	VBS
LRECL:	32 756
BLKSIZE:	6 233

If you want to create the data set manually, it should have a variable record format and a record length (LRECL) of at least 4 092.

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

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

After specifying the required criteria on 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 on 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 symbol (+) and pressing **F4** (Prompt).

Displaying Trace Status and Messages

Use the following collect report data panels to view the status of a collect task in detail and any messages issued by that task.

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

The Trace Status Summary window is shown in Figure 10.

```
DGOMAP10                    Trace Status Summary      Row 1 to 10 of 10

_  Display Status Detail
_  Display messages for this task

Task Description . . . . : Test case for buffer overruns
Data Set Name  . . . . . : 'USERT01.OM.TRACE'
Data Set Status  . . . . : OPEN
DB2 Trace Data Started . : 01/22/98 15:45:25.926
Records Read . . . . . . : 960

Active Traces for this Destination
DSNW127I ¬ CURRENT TRACE ACTIVITY IS -
TNO TYPE   CLASS        DEST QUAL
04  STAT   03           OP2  NO
05  PERFM  04,06,07,17  OP2  NO
*********END OF DISPLAY TRACE SUMMARY DATA*********
DSNW143I ¬ CURRENT TRACE QUALIFICATIONS ARE -
TNO AUTHID   PLAN     RMID         LOCATION
04  *        *        *
05  *        *        *
******END OF DISPLAY TRACE QUALIFICATION DATA******
-- End of List --

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

Figure 10. Trace Status Summary Window

From this window, you can select the *Display Status Detail* field to display the Trace Status Detail window, where you can view further details on 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 the active DB2 traces started by the task, and shows the trace type, class, output buffer destination, and qualification criteria for each active DB2 trace started by the task. If there are many DB2 traces listed, you can use the scrolling keys **F7** (Up) and **F8** (Down) to browse the list.

DB2 Trace Termination

If any collect tasks are active when you exit the Online Monitor, you are notified by one of the Asynchronous Task Termination panels. You can either exit the 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.

DB2 Instrumentation Data

When you collect data using the DB2 Tracing panel or the DB2 START TRACE command, you need to know what traces, classes, and IFCIDs to collect.

The DB2 performance data contains a trace record for each DB2 event. The trace records are identified by an IFCID (instrumentation facility component identifier). In order to have these IFCIDs activated, the appropriate trace types must be started.

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

In some cases, as in accounting trace, most classes contain the same IFCIDs, but the class defines what kind of information is collected from that IFCID.

The instrumentation data types used as input to DB2 PM reporting facility are:

- *Statistics* data shows how much the DB2 system services and database services are used. This information can be used to plan DB2 capacity and to tune an entire set of DB2 programs. In addition, the statistics data contains information about deadlocks, timeouts, and DDF exception events.

  The statistics trace is written at intervals. You can control how often the statistics records are produced.
- *Accounting* data provides information related to application programs.

  The DB2 accounting trace begins collecting this data at successful thread allocation to DB2 and writes a completed record when the thread terminates, is reused, or, if it is a DBAT, when it becomes inactive. Use the accounting data to perform program-related tuning and assess and charge DB2 costs. Accounting data also contains information about packages.
- *Performance* data provides information about a variety of DB2 events. You can use this information to further identify a suspected problem, or to tune DB2 programs and resources for individual users or for DB2 as a whole.

  It is recommended that performance trace be turned on only if there is a specific need for it because some of the performance trace classes have a significant performance overhead.
- *Audit* data provides information about DB2 security controls. It is used to ensure that data access is allowed only for authorized purposes.

The only auditable objects are tables. To audit a table, include the audit clause in the create table or alter table statement. Auditing is not available if this clause is omitted.

- *Global* data is intended primarily for servicing DB2. This data can be externalized using the DB2 PM record trace.

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

The following table lists trace types, classes, and IFCIDs relevant to DB2 PM reports. Refer to it to see what DB2 trace data is used as input to each DB2 PM report set.

Table 1. Input to DB2 PM Reports

DB2 PM Report Set	DB2 Trace Type	Cls	Description of Class	IFCIDs Used by DB2 PM
Accounting	Accounting	1	Accounting data	3, 239
		2	In DB2 time	Additional information for 3
		3	Wait time in DB2	Additional information for 3
		5	Time spent processing IFI requests	Additional information for 3
		7	Package information - in DB2 time	Additional information for 3, 239
		8	Package information - wait time in DB2	Additional information for 3, 239
Audit	Audit	1	Authorization failures	140
		2	Explicit GRANT or REVOKE	141
		3	CREATE, ALTER, and DROP operations against audited tables	105, 107, 142
		4	First change of audited object	105, 107, 143
		5	First read of audited object	105, 107, 144
		6	SQL statement at bind	105, 107, 145
		7	Change in authorization for audited object	55, 83, 87, 169, 312
		8	Utility access to any object	24, 105, 107
I/O activity	Performance	4	Buffer manager I/O and EDM pool requests	6, 7, 8, 9, 10, 29, 30, 105, 107
		5	Log manager	34, 35, 36, 37, 38, 39, 40, 41, 114, 115, 116, 119, 120
		21	Data sharing	105, 107, 255

Table 1. Input to DB2 PM Reports (continued)

DB2 PM Report Set	DB2 Trace Type	Cls	Description of Class	IFCIDs Used by DB2 PM
Locking	Statistics	3	Deadlock and timeout information	172, 196
	Performance	4	Buffer manager I/O and EDM pool requests	105, 107, 226, 227
		6	Locking information	20, 44, 45, 105, 107, 172, 196, 213, 214, 218
		7	Detailed locking information	21, 105, 107, 223
		17	Drain and claim	211, 212, 213, 214, 215, 216
		20	Data sharing	251, 257
		21	Data sharing	259
Record trace	All	All	All traces, classes, and IFCIDs can be used as input	All
SQL activity	Accounting	1	Accounting data	3
		2	In DB2 time	3
		3	Suspensions	3
		5	IFI and data capture events	3
		7	Package information - in DB2 time	239
		8	Package information - wait time in DB2	239
	Performance	2	Subsystem-related events	68, 69, 70, 71, 72, 73, 74, 75, 84, 85, 86, 87, 88, 89, 106, 174, 175
		3	SQL-related events	22, 53, 55, 58, 59, 60, 61, 62, 63, 64, 65, 66, 92, 95, 96, 97, 177, 233, 237, 272, 273
		4	Buffer manager I/O and EDM pool requests	6, 7, 8, 9, 226, 227
		6	Locking information	20, 44, 45, 213, 214, 218
		8	Data manager detail	15, 16, 17, 18, 106, 125, 221, 222, 231, 305, 325
		9	Sort detail	28, 95, 96
		10	Autobind	105, 106, 107, 108, 109
		13	Edit and validation exits	11, 12, 19
		16	Distributed activity	157, 159, 160, 162, 163, 183
		17	Drain and claim detail	213, 214, 215, 216
		30 31 32	Installation-defined classes	188, 324
Statistics	Statistics	1	Statistics data	1, 2
System para-meters	Performance	Any	These IFCIDs are available in all trace classes	106, 201, 202, 256
	Statistics	5	Data sharing global information	230

Table 1. Input to DB2 PM Reports (continued)

DB2 PM Report Set	DB2 Trace Type	Cls	Description of Class	IFCIDs Used by DB2 PM
Utility	Accounting	1	Accounting data	3
	Performance	3	SQL-related events	22, 63, 177
		4	Buffer manager I/O and EDM pool requests	6, 7, 8, 9, 226, 227
		6	Locking information	20, 44, 45, 213, 214, 218
		10	Bind and utilities	23, 24, 25, 105, 107, 108, 109, 110, 111
		13	Edit and validation exits	11, 12, 19
		16	Distributed activity	183
		17	Drain and claim detail	213, 214, 215, 216

DB2 PM gathers input to explain by connecting to an active DB2 subsystem.

If you want to see a count of the input trace records used in a DB2 PM job, you can produce an IFCID frequency distribution log. For more information, refer to "Chapter 7. Generating Commonly Used Reports" on page 61.

Other Ways to Get Trace Data

There are two other methods of getting DB2 trace data for input to DB2 PM reports or traces:
- Using DB2 INSTALL parameters
- Using the DB2 START TRACE command.

INSTALL Parameter Values

Accounting, statistics, and audit traces can be set to start automatically using DB2 INSTALL parameter values.

You can modify the parameters on 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.

On the Tracing panel specify the traces you want to start automatically. You can also specify which classes within the traces you want to activate. All data is sent to the SMF service program by default.

Refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for more information about the INSTALL parameters.

Using DB2 START TRACE Command

All types of DB2 trace data can be obtained by issuing a DB2 START TRACE command.

Enter the command from an OS/390 console, the DSN command processor, the DB2I commands panel, from an IMS or CICS terminal, or the DB2 PM Online Monitor. You must have TRACE privilege or SYSOPR, SYSCTRL, or SYSADM authority to issue the command.

Collecting Data

The following diagram shows the START TRACE command followed by a short description of the parameters and options that you have to specify to get appropriate trace data for analysis with DB2 PM. For a complete description of the START TRACE command refer to the *DB2 Universal Database Server for OS/390 Command Reference*.

```
►►──START TRACE──(──┬─P──┬──)──DEST──(──┬─GTF─┬──)──CLASS──(──┬─*─────────────┬──)──────────────►
                    ├─A──┤                 ├─SMF─┤                │ ┌─,───────┐  │
                    ├─S──┤                 ├─OPn─┤                │ ▼         │  │
                    ├─AU─┤                 └─OPX─┘                └───integer─┴──┘
                    └─G──┘

►─IFCID──(──┬─*─────────────┬──)──AUTHID──(──┬─*────────────┬──)──PLAN──(──┬─*───────────────┬──)──►
            │ ┌─,───────┐   │                │ ┌─,──────┐   │              │ ┌─,──────────┐  │
            │ ▼         │   │                │ ▼        │   │              │ ▼            │  │
            └───integer─┴───┘                └───authid─┴───┘              └───plan-name──┴──┘

►─TDATA──(──┬─COR──┬──)──LOCATION──(──┬─*───────────────────┬──)──────────────────────────────►◄
            ├─CPU──┤                  │ ┌─,───────────────┐ │
            └─DIST─┘                  │ ▼                 │ │
                                      └───location name───┴─┘
```

Figure 11. DB2 START TRACE Command for DB2 PM Processing

The following options can be specified in the START TRACE command:

DEST Destination specifies the buffer where the trace data is directed. The destination can be SMF, GTF, or an OP buffer.

Both SMF and GTF are service programs that provide a means of recording performance data. SMF is usually used for daily monitoring, GTF for monitoring a specific problem when the volume of data is large.

SMF

SMF is used for continuous monitoring. It is the default destination for statistics, accounting, and audit traces.

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.

SMF must be active before it can collect data for DB2. To trace using SMF as the destination:

1. Ensure that SMF is collecting the following records:
   - DB2 accounting records are collected with SMF type 101 records.
   - DB2 audit records are collected with SMF type 102 records.
   - DB2 statistics records are collected with SMF type 100 records.
   - DB2 performance records are collected with SMF type 102 records.

   For additional information about how to activate SMF, refer to the *OS/390 MVS System Management Facilities (SMF)*.
2. Ensure that the SMF data sets and buffers are large enough to hold the data being collected.

   Data sent from DB2 to SMF can get lost. The most common reason is that SMF buffers run out of space. SMF rejects any records sent to it while the shortage of space exists.

   Lost data cannot be recaptured by DB2 PM, but you can produce a long statistics report or trace to show the number of records that have been lost (block DB2 IFC Destination Data).
3. Start the appropriate DB2 trace with DEST(SMF).
4. Trace the DB2 events you are interested in.
5. Stop the DB2 trace.
6. Obtain the data from the SMF data sets to be used by DB2 PM.

   It is highly recommended, although not required, that you wait until the SMF data set has been closed before you attempt to obtain the SMF data to be used by DB2 PM.
7. Run DB2 PM for the appropriate report set against the collected SMF data.

GTF

If you plan to collect a large volume of DB2 trace data, as in the case of monitoring detailed locking activity, you might want to use GTF. You can also use GTF to collect DB2 trace data for a particular problem, or to isolate particular trace data quickly. GTF is the default destination for performance trace.

GTF must be active before it can collect trace data for DB2. To trace using GTF as the destination:

1. Start GTF. It is highly recommended that `TIME=YES` be specified when starting GTF.
2. Start the appropriate DB2 trace with DEST(GTF).
3. Trace the DB2 events you are interested in.
4. Stop the DB2 trace.
5. Stop GTF.
6. Run DB2 PM for the appropriate report set against the collected GTF data.

DB2 records can exceed the record length provided by GTF (256 characters). When this occurs, GTF stores this information in record segments. DB2 PM processes these segments and reconstructs the DB2 trace data into one logical record when it produces the DPMOUT data set, and when it performs report processing.

If segments of DB2 records are missing from the GTF data set, a message is printed in the DB2 PM Job Summary Log.

Records needed by DB2 PM can be lost if GTF is started with the JOBNAMEP option. For example, information related to DBID and OBID translation is not recorded if you use this option. For more information, refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide*.

GTF data can be lost if GTF is not active before DB2 trace is started, or while GTF switches to another device.

Lost data cannot be recaptured by DB2 PM, but you can produce a long statistics report or trace to show the number of records that have been lost (block DB2 IFC Destination Data).

CLASS

Class defines a certain group of data within the trace type. Specify the numbers of the classes you want to activate.

IFCID

You can trace individual IFCIDs to limit the amount of data. This can be helpful in reducing the overhead of running particular classes of DB2 trace data. However, many DB2 PM reports have dependencies on several IFCID combinations, and if the proper records are not generated by the DB2 trace facility, DB2 PM cannot make the proper record relationships to produce a meaningful report.

Note that sometimes the same IFCID is available in different types and classes. For example, IFCID 172 is available in statistics class 3 and performance class 6.

AUTHID and PLAN

Specifying the authorization ID and plan name on the START TRACE command can control the amount of data produced on the SMF/GTF data sets. Choose AUTHID and PLAN parameters carefully, based upon the application and users of the application being monitored.

Consider how you want to use DB2 PM. If you want to examine a specific problem, carefully choose your AUTHID and PLAN to obtain data for the unique problem. If you are doing daily monitoring, select all AUTHIDs and PLANs to get an overview of the activity on your system.

TDATA

The headers relevant to DB2 PM are:

COR To obtain the Correlation Header. The DB2 Correlation Header contains the primary authorization ID, the connection ID, the correlation ID, the plan name and the original authorization ID.

If the Correlation Header is required by a DB2 PM report set, any records without the Correlation Header are not reported.

CPU To obtain the CPU Header. The CPU Header contains the CPU time.

The following DB2 PM report sets use the Correlation and CPU Headers:

Report Set	Correlation Header	CPU Header
Accounting	Required	Not used
Audit	Required	Not used
I/O activity	Required	Not used
Locking	Required	Not used
Record trace	Used if present	Used if present
SQL activity	Highly recommended	Recommended
Statistics	Not used	Not used
Utility activity	Required	Recommended

DIST To obtain the Distributed Header. The Distributed Header contains the requesting location name, requester timestamp, application requester name, and product ID.

When the data contains distributed activity, the Distributed Header is required to obtain that data. The absence of the Distributed Header in this case can considerably alter the content of reports and traces. See the individual report set sections for information about distributed activity within specific reports and traces.

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 of them.

LOCATION

The LOCATION parameter is used to trace data with a distributed relationship for specific locations. Using the LOCATION parameter omits allied threads from the trace. Only allied-distributed threads and DBATs associated with the locations specified are traced.

You can specify up to eight locations; a separate trace is started for each one. 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.

If you omit the LOCATION option, all threads (including allied threads) from all locations are traced. LOCATION has no effect when accounting class 2 is started.

To see what DB2 traces were active during the reporting period, produce a job summary log. For more information, refer to “Chapter 7. Generating Commonly Used Reports” on page 61.

START TRACE Command Examples

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

If you do not specify class, the default 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 type, you must specify one of 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 as well as IFCID 68 and 69 data, specify:

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

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

If you want to start all classes of accounting data, you can either use an asterisk or specify all 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 of them, as in the following example:

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

Chapter 6. Running DB2 PM Jobs

To generate DB2 PM reports, you need to specify the appropriate DB2 PM commands, as well as the ddnames for the required data sets, and submit the job.

Accessing DB2 PM

You can choose between two ways of interacting with DB2 PM:
- Using the DB2 PM Interactive Report Facility (IRF)
- Using the ISPF/PDF editor to enter the JCL and commands.

Interactive Report Facility

The IRF is a menu-driven means of using DB2 PM. You do not need to know the DB2 PM command language to use it. If you have questions, online help is available.

To produce reports and traces using the IRF:
1. Select the DB2 PM reporting functions.
2. Specify the required DD statements.
3. Specify the job statement.
4. Submit the job stream for execution.

You can display, edit, and store the generated job stream and execute it in either batch or in foreground.

ISPF/PDF Editor

If you are familiar with DB2 PM command language, you can use the ISPF/PDF editor to specify the commands and the JCL required to produce the reports and submit the job.

DB2 PM Commands

To produce DB2 PM reports, you need to specify the appropriate commands:
- If you want to streamline DB2 PM processing by reducing the amount of data that needs to be processed, start by specifying the GLOBAL command. This is especially important if your input data set contains data for several days. See “Auxiliary Commands” on page 49.
- Next, specify the report set command to indicate the area of performance you want reported.
- Lastly, specify the subcommands and options to control how you want the data to be reported.

Report Set Commands

The report set command specifies the report set you want to use. Refer to “Report Sets” on page 11 for an overview of the different report sets. The following table shows the report set commands:

Table 2. Report Sets and Report Set Commands

Report Set	Command
Statistics report set	STATISTICS
Accounting report set	ACCOUNTING

Table 2. Report Sets and Report Set Commands (continued)

Report Set	Command
Explain report set	EXPLAIN
SQL activity report set	SQLACTIVITY
Utility activity report set	UTILITY
Locking report set	LOCKING
I/O activity report set	IOACTIVITY
Record trace report set	RECTRACE
Audit report set	AUDIT

All of the report set commands have default subcommands, so often all you need to do is to specify the report set command. The default is always the most commonly used and usually the shortest report in the report set.

Note that there is no command to produce the system parameters report; it is produced automatically when you specify data set information for ddname SYSPRMDD.

Subcommands and Options

Use the subcommands to specify how you want the data to be presented.

REPORT Use this command to generate reports. In reports, data is summarized by DB2 PM identifiers, such as the primary authorization ID or the plan name.

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

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

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

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

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

REDUCE Use this command to aggregate statistics and accounting DB2 events. REDUCE consolidates DB2 events with the same DB2 PM identifiers into one. The reduced data can be saved using the SAVE command.

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

Unless SAVE or processing by INTERVAL is required, you can omit REDUCE.

SAVE
Use this command to save reduced data. You can use the saved data in later reporting and for producing graphs. 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 command to include previously saved data.

Auxiliary Commands

There are five auxiliary commands that help you streamline DB2 PM processing:

CASE
Use the CASE command to accept entries in uppercase or lowercase characters. Specify CASE (SENSITIVE) before any other command if you want that the following commands differenciate between uppercase and lowercase entries. If no CASE command is specified or if you specify CASE (ANY) then lowercase characters are translated to uppercase characters.

GLOBAL
Use this command to identify, for example, the users, plans, or the period in time you want to investigate. To do this:

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

  These values are used as defaults in the subcommands.

DISTRIBUTE
Use this command to produce input data for frequency distribution graphs. See “Frequency Distribution Graphs” on page 166.

GROUP
Use this command to define a group of DB2 PM identifier values for use in reporting. See “Group Data” on page 109.

LIST
Use this command to define a list of DB2 PM identifier values for use in reporting. See “Use Lists” on page 110.

DDNAMEs for DB2 PM Data Sets

This section describes the ddnames for the DB2 PM data sets. Some of them are needed in general DB2 PM processing, others are specific to a report set.

DB2 PM General Data Sets

There are some DB2 PM general data sets that must always be specified, some that need to be specified only if you want specific processing or output, and some are automatically allocated if a particular function needs them.

The following list shows ddnames for the required data sets. Note that *STEPLIB* and *SYSIN* are already defined in the IRF.

STEPLIB
The data set that contains DB2 PM programs.

SYSIN
The data set that contains DB2 PM commands.

INPUTDD
The input data set that contains the DB2 instrumentation data.

DPMLOG
The data set where processing messages are written.

SYSOUT The data set where messages about sorting are written.

The ddnames for the optional data sets are:

DPMPARMS The data set that contains information about changes you have made to DB2 PM 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, refer to “Chapter 10. Customizing DB2 PM Functions” on page 113.

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

JOBSUMDD A data set where information about DB2 PM processing is written. It contains the IFCID frequency distribution log and the job summary log.

DISTDD The data set where the output from the DISTRIBUTE command is written. See “Chapter 13. Producing Graphs” on page 157 for information about this command.

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, refer to “Chapter 8. Exception Reporting” on page 95.

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 is written.

EXFILDD1 The data set where the exception log file data set is written.

DB2 PM Report Data Sets

Output from DB2 PM 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 prefixes are:

Accounting	Audit	I/O Activity	Locking	Record Trace	SQL Activity	Statistics	Utility Activity
AC	AU	IO	LO	RT	SQ	ST	UT

The following list shows the default ddnames for the report data sets. *xx* stands for the prefix.

xxRPTDD The data set where report output is written.

xxTRCDD1 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 xxTRCDD2, the third to xxTRCDD3, the fourth to xxTRCDD4, and the fifth to xxTRCDD5.

xxFILDD1 The data set where output from the FILE command is written.

xxSAVDD The data set where data is stored using the SAVE command.

xxRSTDD The data set from where data is read using the RESTORE command.

xxWORK The data set where the output from REDUCE is written. Normally this is a temporary data set that DB2 PM automatically creates and deletes. Only specify a ddname for this data set if you want to control its placement or size.

As the explain and system parameters report setsdo not use subcommands, you only need to specify one data set for each of them. The default ddnames for these data sets are:

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

SYSPRMDD The data set where information about DB2 system parameters is written.

The save, distribute, and job summary data sets are VSAM data sets. For more information, refer to “Appendix A. DB2 PM VSAM Data Sets” on page 271.

Using the IRF to Create Reports

The Interactive Report Facility (IRF) provides the means to generate DB2 PM function requests interactively. Using IRF, you can request all report sets and all functions.

This section provides only a short overview of the available IRF functions. For more details, look at the online help panels. You can invoke them with **F1**.

The IRF is accessed by selecting option 1 (*Create and execute DB2 PM commands*) from the DB2 PM main menu as shown in Figure 12 on page 54.

Using the Interactive Report Selections panel as shown in Figure 13 on page 55 and its associated panels, you can:
- Select report set and processing options
- View command selections
- Delete command selections
- Save command selections for future use
- Recall and modify previously saved selections
- Execute the job in foreground
- Generate the command stream and JCL to execute the job in background.

Selecting Commands and Options

The Interactive Report Selections panel contains several input fields. These input fields are:

- Report set function selection fields

  The Interactive Report Selections panel provides a matrix for specifying the report set (as listed vertically) and the function required (as shown horizontally). You can specify a combination of the report set and function by typing a slash (/), as shown in Figure 14 on page 55, at the point where a row meets a column.

A greater-than symbol (>) indicates a previously made selection. To revoke a previously made selection, erase the greater-than symbol by entering a blank.

- Additional function selection fields

  Beneath the report set function selection fields are selection fields for *Global Processing*, *Frequency Distribution*, *System Parameters*, *Exception log*, and *Explain*. These functions can be requested in the same way as the report set function selections. They are described in detail in the *DB2 PM Report Reference*.

To facilitate and complete the generation of DB2 PM function requests, you can use the following commands:

- INCLUDE/EXCLUDE

  This command is used to filter data during DB2 PM processing. INCLUDE and EXCLUDE can be specified on a report set level or at a global level.
- GROUP/LIST (during INCLUDE/EXCLUDE processing)

  The GROUP command lets you define a named group of DB2 PM identifier values and use the group name when you request reports.

  The LIST command lets you define a named list of values for a DB2 PM identifier, and use the list name in INCLUDE or EXCLUDE instead of individually entering each list member.
- BROWSE

  You can use BROWSE to review the DB2 PM command stream that was generated based on your current selections.
- SAVE

  Use the SAVE command to save current command selections.
- RECALL

  Use the RECALL command to recall previously saved command selection criteria.
- OPTIONS

  Use the OPTIONS command to change the options of your IRF session.
- RESET

  Use the RESET command to clear the input fields.

Executing the Job

Once you have entered your selections, all you need to do is press **F5** (Compose) to generate the jobs that produce the reports.

If you have not specified defaults using the OPTIONS command, the Execution Mode window is displayed for selecting foreground or background processing.

Depending on which execution mode you choose, either the Foreground or the Background DDname Selections panel is displayed. The Background DDname Selections panel is shown in Figure 18 on page 58. The Foreground DDname Selections panel looks similar.

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 the output is directed to. If the name is not enclosed in apostrophes, your 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 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 with an asterisk (*) in the *Required* field.

Press **Enter** to validate the entries, then press **Enter** again to continue. If you choose foreground processing, the job is executed immediately. Your terminal remains busy until the job is completed. If you choose background processing, the Job Processing Selections panel is displayed, as shown in Figure 19 on page 58.

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

To select any of these options, enter the number that identifies the option in the input field. Before you submit the job, you must also type a job statement that is required for your installation on the *Job statement information* lines. To submit the job, type `4` in the input field and press **Enter**.

Saving and Recalling Selections

To facilitate working with IRF, you can use the SAVE and RECALL commands. They are helpful when it comes to saving the selections you have made and recalling the saved selections when preparing a similar job stream.

The SAVE command saves all reporting command, ddname, and report set selections you have made into a partitioned data set. You can also save incomplete selections and use them 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 wish 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, just 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 request a short accounting report.

1. Start IRF. The DB2 PM main menu is displayed.
2. Type `OPTIONS` on the command line. The Session Options window is displayed.
3. Type `1` in the option field for *Execution mode* to select background processing and press **Enter**. The DB2 PM main menu is redisplayed.

```
DGOFMENU           IBM Database 2 Performance Monitor

Select one of the following.

1_ 1. Create and execute DB2 PM commands
   2. Display and print graphs
   3. View online DB2 activity
   4. Maintain parameter data sets
   5. Customize DB2 PM report and trace layouts
   6. Exception profiling



 IBM DB2 UDB Performance Monitor for OS/390 V6
 Licensed Materials - Property of IBM
 5645-DB2 (C) Copyright IBM Corp. 1985, 1998.
 All rights reserved.
 US Government Users Restricted Rights -
 Use, duplication or disclosure restricted
 by GSA ADP Schedule Contract with IBM Corp.

Command ===> __________________________________________________________________
 F1=Help     F2=Split    F3=Exit     F6=History  F9=Swap     F12=Cancel
F17=Collect
```

Figure 12. DB2 PM Main Menu

4. Select option 1 (*Create and execute DB2 PM commands*), as shown in Figure 12 and press **Enter**.

   The Interactive Report Selections menu is displayed, as shown in Figure 13. The first time you use the IRF, a greater-than sign (>) is displayed in the selection fields for *Accounting Report* and *Statistics Report*.

```
DGOOMENU                 Interactive Report Selections

Select functions as required, then press Enter.

Report Set              ---------------  Function  ---------------
                        Reduce Report Trace  File    Save   Restore
Accounting . . . . . . _      >      _      _       _      _
Statistics . . . . . . _      >      _      _       _      _
SQL Activity . . . . . _      _      _
Locking  . . . . . . . _      _      _      _
I/O Activity . . . . . _      _
Audit  . . . . . . . . _      _      _      _
Utility  . . . . . . . _      _      _
Record Trace . . . . .               _      _

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   F13=Recall   F14=Save
```

Figure 13. Interactive Report Selections Panel

5. If you want to delete all previously made selections, use the RESET command. The RESET command also clears the input fields on the panel. Then, on the *Accounting* line, specify *Report* by entering a slash (/). If you do not want to lose your previous selections, do not use the RESET command, but simply overtype the > sign with a / as shown in Figure 14.

```
DGOOMENU                 Interactive Report Selections

Select functions as required, then press Enter.

Report Set              ---------------  Function  ---------------
                        Reduce Report Trace  File    Save   Restore
Accounting . . . . . . _      /      _      _       _      _
Statistics . . . . . . _      >      _      _       _      _
SQL Activity . . . . . _      _      _
Locking  . . . . . . . _      _      _      _
I/O Activity . . . . . _      _
Audit  . . . . . . . . _      _      _      _
Utility  . . . . . . . _      _      _
Record Trace . . . . .               _      _

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   F13=Recall   F14=Save
```

Figure 14. Selecting the Accounting Report

6. Press **Enter**. The Accounting REPORT Selections panel is displayed.

```
DGOOASRE              Accounting REPORT Selections

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

   DDname    User Comment
/  ________  Example accounting report_____
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________
_  ________  ______________________________

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

Figure 15. Accounting REPORT Selections Panel

7. Type a slash (/) in the action field, and `Example accounting report` in the *User Comment* field, as shown in Figure 15.

   You need not specify a ddname; the default is used.
8. Press **Enter**. The Accounting REPORT panel is displayed.

```
DGOOAREP                   Accounting REPORT

Update fields as required, then press Enter.

                                                               More:     +
User comment  . . . . . . . Example accounting report
DDname  . . . . . . . . . . ACRPTDD

Layout  . . . . . . . . . . SHORT_  User-tailored Report Format

Exception . . . . . . . . . _  1=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 16. Accounting Report Panel

9. Type `SHORT` in the *Layout* field to use the short sample layout. If you wanted to use a user-tailored report, you would specify its name here.
10. Press **Enter** and **F3** to return to the Interactive Report Selections panel.

Now you have completed the specification for the DB2 PM command, subcommands, and options required to generate the example accounting report.

11. To view the generated DB2 PM command stream, press **F6** (Browse). The DB2 PM command stream is shown in Figure 17.

```
 DGOFBRWS FPB.SPFTEMP1.CNTL -------------------------- Line 00000000 Col 001 080
********************************* TOP OF DATA **********************************
ACCOUNTING
            REPORT                    /*Example accounting report*/
                     DDNAME(ACRPTDD)
                     LAYOUT(SHORT)
EXEC
******************************** BOTTOM OF DATA *******************************









Command ===> ________________________________________________ Scroll ===> CSR_
 F1=Help    F2=Split   F3=Exit    F5=Rfind   F7=Up      F8=Down    F9=Swap
F10=Left   F11=Right  F12=Cancel
```

Figure 17. Browsing the DB2 PM Command Stream

12. Press **F3** or **F12** (Cancel) to return to the Interactive Report Selections panel.
13. From the Interactive Report Selections panel, press **F5** (Compose). Compose generates the JCL and command stream, which you can browse, edit, store, or execute. The Background DDname Selections panel is displayed next, as shown in Figure 18.
14. Use the Background DDname Selections panel to specify the data set information for the required system and report data sets. The DGO.V6R1M0.SDGODATA(DGOCIVPI) data set is shipped as part of DB2 PM, so you can use it if you do not have other input data available.

    Note that if you do not specify data set information for the accounting report ddname (ACRPTDD) or the statistics report ddname (STRPTDD), these data sets are allocated dynamically. The ddname for the job summary log (DPMLOG) is also allocated dynamically, and it is produced every time you run a DB2 PM job.

```
DGOOJOBQ              Background DDname Selections

Update the data set information.  Select one or more data sets to add
extended information, then press Enter.  Your TSO prefix is added to data
set names not enclosed in single quotes.  Press Enter to continue when
complete.


      DDname       Data Set Information                              Required

  _   INPUTDD      'DGO.V6R1M0.SDGODATA(DGOCIVPI)'_______________   *
  _   EXCPTDD      ______________________________________________
  _   EXTRCDD1     ______________________________________________
  _   EXFILDD1     ______________________________________________
  _   ACRPTDD      ______________________________________________
  _   DPMLOG       ______________________________________________
  _   DPMOUTDD     ______________________________________________
  _   DPMPARMS     ______________________________________________
  _   JOBSUMDD     ______________________________________________
  _   JSSRSDD      ______________________________________________
  _   STRPTDD      ______________________________________________
  _   SYSOUT       ______________________________________________
  _   SYSUDUMP     ______________________________________________
  ******************************* Bottom of data ****************************
Command ===> ______________________________________________________________
 F1=Help    F2=Split   F3=Exit    F7=Up      F8=Down    F9=Swap   F12=Cancel
```

Figure 18. Background DDname Selections Panel

15. After you have completed the data set information, press **Enter** to validate the entries.
16. Press **Enter** again. The Job Processing Selections panel is displayed, as shown in Figure 19.

```
DGOOJOBM                  Job Processing Selections

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
//USERPMA JOB (D01,CHAT),'DB2 PM USER',MSGCLASS=V,CLASS=D,_____________
//             REGION=0M,NOTIFY=USERPM__________________________________
_______________________________________________________________________
_______________________________________________________________________
_______________________________________________________________________









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

Figure 19. Job Processing Selections Panel

17. To submit the job, select option 4, type the appropriate job statement on the *Job statement information* lines, and press **Enter**. The job statement shown in Figure 19 is just an example. A message is displayed to indicate whether the submit has been successful or not.

Using the ISPF/PDF Editor

If you are familiar with DB2 PM, you can enter the ddname specifications and DB2 PM commands using the ISPF/PDF editor.

To produce the same accounting and statistics reports as in the previous example (layout SHORT), you need to enter a job statement, the ddnames, and DB2 PM commands shown in Figure 20.

```
EDIT ---- SYS92226.T092210.RA000.USERPMA.R000003 ------------ Columns 001 072
Command ===> _________________________________________________ Scroll ===> CSR_
****** ***************************** TOP OF DATA ******************************
000001 //USERPMA JOB  (TTS1,YUS7),'DB2PM-IBM',
000002 //             MSGCLASS=V,CLASS=D,NOTIFY=USERPM
000003 //        EXEC PGM=DB2PM
000004 //STEPLIB  DD  DSN=DGO.V6R1M0.SDGOLOAD,DISP=SHR
000005 //INPUTDD  DD  DSN=DGO.V6R1M0.SDGODATA(DGOCIVPI),DISP=SHR
000010 //JOBSUMDD DD  SYSOUT=A
000011 //SYSIN    DD  *
000012 ACCOUNTING REPORT
000013 STATISTICS REPORT
000014 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 20. Specifying a Job Stream Using ISPF/PDF Editor

The job statement, ddnames, and DB2 PM commands that must be specified are highlighted. JOBSUMDD is optional, but it is good practice to include it.

In the above example REPORT has been specified only as an illustration; it could just as well have been left out because REPORT is the default for both accounting and statistics.

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

```
EDIT ---- SYS92226.T092210.RA000.USERPMPA.R0000003 ---------- Columns 001 072
Command ===> SUBMIT___________________________________________ Scroll ===> CSR_
****** ***************************** TOP OF DATA ******************************
000001 //USERPMA JOB (TTS1,YUS7),'DB2PM-IBM',
⋮
```

Figure 21. Submitting the Job

Output

Figure 22 shows the accounting report produced in the previous examples.

```
   LOCATION: STLEC1                                DB2 PERFORMANCE MONITOR (V6)                                PAGE: 1-1
      GROUP: DSNCAT                                   ACCOUNTING REPORT - SHORT                      REQUESTED FROM: NOT SPECIFIED
     MEMBER: SSDQ                                                                                                TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                     ORDER: PRIMAUTH-PLANNAME                        INTERVAL FROM: 04/06/99 20:18:00.23
DB2 VERSION: V6                                           SCOPE: MEMBER                                          TO: 04/06/99 20:48:38.68

PRIMAUTH                    #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
 PLANNAME                   #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
--------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

ADMF001                           9       0    0.00    4.22    1.00    0.00      30.714273      30.067720   368.22    99.89     0.00
 DSNTEP61                         0      15    3.67    0.56    0.56   10.22       0.346763       0.305899   103.11     1.11        0

ADMF001                          22       3    0.00    0.00    0.00    0.00       8.761319       7.751928    32.95     3.95     0.00
 DSNUTIL                          0     113    0.00    0.00    0.00    0.00       0.093037       0.070805    15.68     0.00        0

***  TOTAL  ***
ADMF001                          50       5    0.00    1.40    0.18    0.06      14.158384      13.567541   184.35    39.78     0.00
                                  0     158    0.66    0.10    0.10    1.88       0.143647       0.126001    64.46     0.62        0

SYSADM                            2       0    0.00    0.00    0.00    0.00      29.882355      29.187787     2.00     0.00     0.50
 DSNTEP61                         0       2    0.00    0.00    0.00    2.00       0.024030       0.009512     0.00     0.00        0

***  GRAND TOTAL  ***
                                 54       5    0.00    1.30    0.17    0.06      14.234724      13.661916   175.00    37.74     0.02
                                  0     162    0.61    0.09    0.09    1.81       0.134015       0.117135    61.15     0.59        0

ACCOUNTING REPORT COMPLETE
```

Figure 22. Accounting Report—Model Layout SHORT

Figure 23 shows the statistics report produced in the previous examples.

```
   LOCATION: STLEC1                                DB2 PERFORMANCE MONITOR (V6)                                PAGE: 1-1
      GROUP: DSNCAT                                   STATISTICS REPORT - SHORT                      REQUESTED FROM: NOT SPECIFIED
     MEMBER: SSDQ                                                                                                TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                                                                     INTERVAL FROM: 04/06/99 20:10:37.48
DB2 VERSION: V6                                           SCOPE: MEMBER                                          TO: 04/06/99 20:37:56.36

---- HIGHLIGHTS ----------------------------------------------------------------------------------------------------------------------
INTERVAL START: 04/06/99 20:10:37.48    INTERVAL ELAPSED: 27:18.88309    INCREMENTAL BINDS       :     0.00    DBAT QUEUED:        N/P
INTERVAL END  : 04/06/99 20:37:56.36    OUTAGE ELAPSED  :    0.000000    AUTH SUCC.W/OUT CATALOG:     11.00    DB2 COMMAND:      51.00
SAMPLING START: 04/06/99 20:10:37.48    TOTAL THREADS   :       38.00    BUFF.UPDT/PAGES WRITTEN:      4.93    TOTAL API  :       0.00
SAMPLING END  : 04/06/99 20:37:56.36    TOTAL COMMITS   :       63.00    PAGES WRITTEN/WRITE I/O:      1.93    MEMBER     :        N/A

CPU TIMES                               TCB TIME         SRB TIME       TOTAL TIME     OPEN/CLOSE ACTIVITY        QUANTITY
-------------------------------  ---------------  ---------------  ---------------     -------------------------  --------
SYSTEM SERVICES ADDRESS SPACE           0.401153         0.537096         0.938249     OPEN DATASETS - HWM           83.00
DATABASE SERVICES ADDRESS SPACE         2.004806         0.527284         2.532090     OPEN DATASETS                 68.00
IRLM                                    0.000413         0.266406         0.266819     IN USE DATA SETS               3.00
DDF ADDRESS SPACE                       0.012715         0.006164         0.018880

SQL DML   QUANTITY    SQL DCL         QUANTITY    SQL DDL     QUANTITY    LOCKING ACTIVITY  QUANTITY    DATA SHARING LOCKS  QUANTITY
--------  --------    --------------  --------    ----------  --------    ----------------  --------    ------------------  --------
SELECT        0.00    LOCK TABLE          0.00    CREATES        16.00    DEADLOCKS             0.00    GBL CONT.RATE (%)       6.46
INSERT       38.00    GRANT               0.00    DROPS           9.00    TIMEOUTS              0.00    FLS CONT.RATE (%)       1.48
UPDATE        9.00    REVOKE              0.00    ALTERS          0.00    SUSPENSIONS-LOCK      0.00    L-LOCKS RATE (%)        0.05
DELETE        0.00    SET HOST VAR.       0.00    RENAME TBL      0.00    SUSPENSIONS-OTHR      0.00    LOCK REQ.(P-LOCK)     296.00
PREPARE      92.00    SET SQLID           0.00    COMMENT ON      0.00    LOCK REQUESTS      3013.00    UNLOCK REQ.(P-LCK)     58.00
⋮

⋮
```

Figure 23. Statistics Report—Model Layout SHORT

See “Part 4. Monitoring and Tuning with DB2 PM Batch” on page 175 for information on how to interpret the DB2 PM reports and monitor and tune DB2.

For a comprehensive description of all the fields in all the report sets, refer to the appropriate part in the *DB2 PM Report Reference*.

Chapter 7. Generating Commonly Used Reports

This chapter describes how to request the most commonly used DB2 PM reports. These are:
- Accounting
- Explain
- System parameters
- Statistics
- SQL activity
- Locking.

It is recommended that you run the accounting, system parameters, and statistics reports regularly; when you discover a performance problem, you might want to run an explain, SQL activity, or locking report, depending on the type of problem.

In addition to the reports, you might want to generate an IFCID frequency distribution log and the job summary log every time you run a DB2 PM job. The IFCID frequency distribution log shows a count of the input and processed trace records, and the job summary log provides information about the job, such as the START TRACE commands detected in the input data.

This chapter describes reports rather than traces, because reports present summarized data and are therefore shorter. The only exception is the SQL activity report set, in which the traces are more straightforward and therefore recommended for the first-time user.

You must have the appropriate input data available before you can run DB2 PM reports. Refer to Table 1 on page 37 for information about the trace classes to start for the different report sets.

This chapter shows how to request the reports using DB2 PM command language.

For descriptions of how to use the information on the reports for tuning DB2, refer to “Part 4. Monitoring and Tuning with DB2 PM Batch” on page 175. For descriptions of all the individual fields on the reports, refer to the *DB2 PM Report Reference*.

Accounting

This section contains information about how to produce accounting reports. Use the accounting reports to examine application performance.

Note that the information included in the accounting reports can be modified. To find out how, refer to “Tailoring Report Layouts” on page 115.

Short Report

The most commonly used accounting report is the short report. It shows summarized data about application activity (times, SQL, and buffer pools activity) and, if available, package or DBRM and DDF data.

To generate this report, all you need to specify is:

```
⋮

ACCOUNTING
⋮
```

In this report, data is by default summarized for every plan within a primary authorization ID.

```
   LOCATION: STLEC1                            DB2 PERFORMANCE MONITOR (V6)                                PAGE: 1-1
      GROUP: DSNCAT                               ACCOUNTING REPORT - SHORT                      REQUESTED FROM: NOT SPECIFIED
     MEMBER: SSDQ                                                                                           TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                  ORDER: PRIMAUTH-PLANNAME                       INTERVAL FROM: 04/06/99 20:18:00.23
DB2 VERSION: V6                                        SCOPE: MEMBER                                         TO: 04/06/99 20:48:38.68

PRIMAUTH                    #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
 PLANNAME                   #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
--------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

ADMF001                          13       0    0.00    0.00    0.00    0.00       0.662349       0.662264      N/P      N/P     0.00
 'BLANK'                          0      13    0.00    0.00    0.00    0.00       0.004677       0.004589      N/P      N/P        0

ADMF001                           3       2    0.00    0.00    0.00    0.00       5.161409       5.161304   220.67     9.00     0.00
 DSNBIND                          0       3    0.00    0.00    0.00    0.00       0.139584       0.139478   157.00     0.67        0
⋮

⋮
```

Figure 24. Short Accounting Report

Ordering the Data

Sometimes you might want to report data by, for example, different plans instead of plans within primary authorization IDs. To do that, specify:

```
⋮

ACCOUNTING
     REPORT
        ORDER (PLANNAME)
⋮
```

The generated report shows the same data as in the previous example, except that it is summarized for every plan present in the input data.

```
   LOCATION: STLEC1                            DB2 PERFORMANCE MONITOR (V6)                                PAGE: 1-1
      GROUP: DSNCAT                               ACCOUNTING REPORT - SHORT                      REQUESTED FROM: NOT SPECIFIED
     MEMBER: SSDQ                                                                                           TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                    ORDER: PLANNAME                              INTERVAL FROM: 04/06/99 20:18:00.23
DB2 VERSION: V6                                        SCOPE: MEMBER                                         TO: 04/06/99 20:48:38.68

                            #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
PLANNAME                    #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
--------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

DSNTEP61                         11       0    0.00    3.45    0.82    0.00      30.563015      29.907732   301.64    81.73     0.09
                                  0      17    3.00    0.45    0.45    8.73       0.288084       0.252011    84.36     0.91        0

DSNUTIL                          22       3    0.00    0.00    0.00    0.00       8.761319       7.751928    32.95     3.95     0.00
                                  0     113    0.00    0.00    0.00    0.00       0.093037       0.070805    15.68     0.00        0

⋮

⋮
```

Figure 25. Short Accounting Report Ordered by Planname

When you want to examine package or DBRM information regardless of the plan to which the packages or DBRMs belong, you can produce a report ordered by PACKAGE.

To produce a short report ordered by PACKAGE, specify:

```
⋮
ACCOUNTING
     REPORT
        ORDER  (PACKAGE)
⋮
```

When a report is ordered by PACKAGE, only package-related or DBRM-related information is printed, because other values cannot be attributed to a particular package or DBRM. Therefore only the information that is usually at the end of a report entry is now printed.

```
   LOCATION: STLEC1                             DB2 PERFORMANCE MONITOR (V6)                              PAGE: 1-1
      GROUP: N/P                                   ACCOUNTING REPORT - SHORT                    REQUESTED FROM: NOT SPECIFIED
     MEMBER: N/P                                                                                            TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                        ORDER: PACKAGE                          INTERVAL FROM: 04/06/99 20:18:00.23
DB2 VERSION: V6                                          SCOPE: MEMBER                                      TO: 04/06/99 20:48:38.68

                                                            TYPE           SQLSTMT   CL7 CPU TIME  CL8 SUSP
PACKAGE                                                      #OCCURS  CL7 ELAP.TIME  CL8 SUSP.TIME
----------------------------------------------------------  --------  -------------  -------------  --------

STLEC1.CLPFR271.PFR27A                                      PACKAGE            1.00       0.001808       N/P
                                                                   1       0.064458            N/P

STLEC1.CLPFR271.PFR27B                                      PACKAGE            1.00       0.002060       N/P
                                                                   1       0.095079            N/P

STLEC1.CLPFR271.PFR27C                                      PACKAGE            1.00       0.003304      1.00
                                                                   1       0.380271       0.000292
⋮

⋮
```

Figure 26. One Entry on a Short Accounting Report Ordered by Package

For more information about reporting packages, refer to “Defining the MAINPACK Identifier” on page 133.

You can also use other DB2 PM identifiers to order the data on your reports. For a description of the most commonly used ones, refer to “DB2 PM Identifiers” on page 107.

TOP Lists

Often it is useful to identify the report entries that might indicate a problem application. The sign of a problem can be a long elapsed, processing or DB2 time, or a high number of suspensions, commits per update, or buffer updates. However, identifying these entries can be difficult when your report covers many users over a long period of time and it is not immediately clear which applications might be causing performance problems.

To identify report entries with a high value in certain fields, you can produce an accounting report with TOP lists, or filter the input data using the TOP ONLY option. The TOP lists indicate which entries on the report have the highest value in the field you have specified using the TOP keyword whereas the TOP ONLY reports and traces only show entries that contain the highest values for the TOP field.

There are many TOP keywords to choose from, such as elapsed, processing, and wait times, a number of different kinds of SQL statements, or package-related or DBRM-related information. By default, the TOP list contains the top ten entries, but you can change the number to anything from one to fifty.

For example, to produce a short accounting report with a list of the top three plans that spent the longest time in DB2, specify:

```
⋮
ACCOUNTING
     REPORT
        ORDER (PLANNAME)
        TOP   (3 INDB2ET)
⋮
```

The generated report is otherwise the same as the previous one, except that the last page shows a list of the three plans that had the highest value for elapsed time in DB2. This list points to the page on which the report entries can be found.

```
   LOCATION: STLEC1                           DB2 PERFORMANCE MONITOR (V6)                          PAGE: 1-2
      GROUP: DSNCAT                              ACCOUNTING REPORT - SHORT                 REQUESTED FROM: NOT SPECIFIED
     MEMBER: SSDQ                                                                                     TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                      ORDER: PLANNAME                      INTERVAL FROM: 04/06/99 20:18:00.23
DB2 VERSION: V6                                        SCOPE: MEMBER                                  TO: 04/06/99 20:48:38.68

TOP FIELD: ELAPSED TIME SPENT IN DB2                                           TOP NUMBER REQUESTED:    3

        PLANNAME                                                     VALUE    PAGE
----    --------------------------------------------------------    ---------------    ----------
   1    FMBVD141                                                1:11.043934    1-1
   2    DSNTEP61                                                  29.907732    1-1
   3    DSNUTIL                                                    7.751928    1-1

ACCOUNTING REPORT COMPLETE
```

Figure 27. TOP List Page of an Accounting Report

If you had specified the following instead:

```
⋮
ACCOUNTING
     REPORT
        ORDER (PLANNAME)
        TOP   (3 ONLY INDB2ET)
⋮
```

your report would only include the three entries for the three plans that have the highest elapsed time in DB2.

Exception Report

Another efficient way to identify potential problem applications is to use exception processing. The exception reports only show report entries with fields containing values outside limits you have specified. For more information about setting the limits, refer to “Chapter 8. Exception Reporting” on page 95.

To generate the exception report, specify:

```
⋮
ACCOUNTING
     REPORT
        EXCEPTION
⋮
```

The exception report is much shorter than the report that is shown in Figure 25 on page 62 , because it only contains entries that have values exceeding the limits you have specified. A block of exception information is printed after every entry.

```
   LOCATION: SANTA_TERESA_LAB                 DB2 PERFORMANCE MONITOR (V6)                           PAGE: 1-1
      GROUP: N/P                                ACCOUNTING REPORT - SHORT                  REQUESTED FROM: NOT SPECIFIED
     MEMBER: N/P                                        EXCEPTION                                      TO: NOT SPECIFIED
  SUBSYSTEM: V41A                                 ORDER: PRIMAUTH-PLANNAME                  INTERVAL FROM: 04/30/99 20:34:09.92
DB2 VERSION: V6                                     SCOPE: MEMBER                                      TO: 04/30/99 20:34:09.92

PRIMAUTH                    #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
 PLANNAME                   #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
--------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

ADMF001                           1       0    0.00    0.00    0.00    0.00       3.613503            N/P    33.00     0.00     0.00
 DSNTEP61                         0       1   21.00    1.00    1.00    2.00       0.289514            N/P     0.00     0.00        0

  *********************************************************************************************************************************
  *  TYPE           FIELD ID   FIELD DESCRIPTION                                   BY                      VALUE   THRESHOLD     *
  *                            FIELD QUALIFIER                                                                                 *
  *  WARNING        QXREDGRP   PARALL.GROUPS RUN WITH REDUCED DEGREE               TOTAL                       0 < 1             *
  *                                                                                                                            *
  *********************************************************************************************************************************
```

Figure 28. Short Accounting Exception Report

Long Report

Sometimes you might want to see more detailed information about application performance than the short accounting report provides. In this case you can generate a long accounting report.

The long report shows information from all accounting categories, including application times, SQL and locking activity, buffer pool information, package or DBRM data, and distributed data if available.

To generate a long report, specify:

```
⋮
ACCOUNTING
     REPORT
        LAYOUT (LONG)
⋮
```

As you can see, one entry on the long report can extend over several pages, whereas on the short report an entry consists of only a couple of lines. The following example shows just one entry on a long accounting report:

```
   LOCATION: STM4D61Y                              DB2 PERFORMANCE MONITOR (V6)                                 PAGE: 1-1
      GROUP: N/P                                     ACCOUNTING REPORT - LONG                     REQUESTED FROM: NOT SPECIFIED
     MEMBER: N/P                                                                                             TO: NOT SPECIFIED
  SUBSYSTEM: Y61Y                                  ORDER: ENDUSER-PRIMAUTH-WSNAME                  INTERVAL FROM: 01/29/99 23:48:01.86
DB2 VERSION: V6                                           SCOPE: MEMBER                                      TO: 01/29/99 23:53:34.20

ENDUSER: HUGO  PRIMAUTH: SOF  WSNAME: WORKSTATNAME

ELAPSED TIME DISTRIBUTION                                          CLASS 2 TIME DISTRIBUTION
----------------------------------------------------------------   ----------------------------------------------------------------
APPL   !> 1%                                                       CPU    !===========> 22%
DB2    !===============================================> 97%       NOTACC !=====================================> 76%
SUSP   !=> 2%                                                      SUSP   !=> 2%

AVERAGE       APPL(CL.1)  DB2 (CL.2)  IFI (CL.5)    CLASS 3 SUSPENSIONS   AVERAGE TIME  AV.EVENT    HIGHLIGHTS
------------  ----------  ----------  ----------    --------------------  ------------  --------    --------------------------
ELAPSED TIME  1:01.61094  1:01.00275         N/P    LOCK/LATCH(DB2+IRLM)      4.143163     31.83    #OCCURRENCES    :        6
 NONNESTED    1:01.61094  1:01.00275         N/A    SYNCHRON. I/O             0.871526    164.33    #ALLIEDS        :        6
 STORED PROC    0.000000    0.000000         N/A     DATABASE I/O             0.871526    164.33    #ALLIEDS DISTRIB:        0
 UDF            0.000000    0.000000         N/A     LOG WRITE I/O            0.000000      0.00    #DBATS          :        0
 TRIGGER        0.000000    0.000000         N/A    OTHER READ I/O            1.654408     96.83    #DBATS DISTRIB. :        0
                                                    OTHER WRTE I/O            0.024326      0.67    #NO PROGRAM DATA:        6
CPU TIME       56.879690   56.838591         N/P    SER.TASK SWTCH            1.666484     20.67    #NORMAL TERMINAT:        6
 AGENT         13.210025   13.168926         N/A     UPDATE COMMIT            0.000000      0.00    #ABNORMAL TERMIN:        0
  NONNESTED    13.210025   13.168926         N/P     OPEN/CLOSE               1.564079     10.33    #CP/X PARALLEL. :        6
  STORED PRC    0.000000    0.000000         N/A     SYSLGRNG REC             0.005893      1.00    #IO PARALLELISM :        0
  UDF           0.000000    0.000000         N/A     EXT/DEL/DEF              0.088636      3.33    #INCREMENT. BIND:        0
  TRIGGER       0.000000    0.000000         N/A     OTHER SERVICE            0.007875      6.00    #COMMITS        :       12
 PAR.TASKS     43.669665   43.669665         N/A    ARC.LOG(QUIES)            0.000000      0.00    #ROLLBACKS      :        0
                                                    ARC.LOG READ              0.000000      0.00    MAX SQL CASC LVL:        0
SUSPEND TIME         N/A    8.359907         N/A    STOR.PRC SCHED            0.000000      0.00    UPDATE/COMMIT   :     0.00
 AGENT               N/A    1.310411         N/A    UDF SCHEDULE              0.000000      0.00    SYNCH I/O AVG.  : 0.005303
 PAR.TASKS           N/A    7.049495         N/A    DRAIN LOCK                0.000000      0.00
                                                    CLAIM RELEASE             0.000000      0.00
NOT ACCOUNT.         N/A   46.523416         N/A    PAGE LATCH                0.000000      0.00
DB2 ENT/EXIT         N/A       19.00         N/A    NOTIFY MSGS               0.000000      0.00
EN/EX-STPROC         N/A        0.00         N/A    GLOBAL CONT.              0.000000      0.00
EN/EX-UDF            N/A        0.00         N/A    FORCE-AT-COMMIT           0.000000      0.00
DCAPT.DESCR.         N/A         N/A         N/P    TOTAL CLASS 3             8.359907    314.33
LOG EXTRACT.         N/A         N/A         N/P

SQL DML   AVERAGE    TOTAL    SQL DCL           TOTAL    SQL DDL    CREATE   DROP  ALTER    LOCKING                AVERAGE    TOTAL
-------- -------- --------    -------------- --------    ---------- ------ ------ ------    --------------------- -------- --------
SELECT       0.00        0    LOCK TABLE            0    TABLE           1      0      0    TIMEOUTS                  0.00        0
INSERT       0.00        0    GRANT                 0    TEMP TABLE      0    N/A    N/A    DEADLOCKS                 0.00        0
UPDATE       0.00        0    REVOKE                0    AUX TABLE       0    N/A    N/A    ESCAL.(SHARED)            0.00        0
DELETE       0.00        0    SET CURR.SQLID        0    INDEX           1      0      0    ESCAL.(EXCLUS)            0.00        0
                              SET HOST VAR.         0    TABLESPACE      1      1      0    MAX PG/ROW LOCKS HELD   102.17      170
DESCRIBE     0.00        0    SET CUR.DEGREE        6    DATABASE        0      0      0    LOCK REQUEST            290.83     1745
DESC.TBL     0.00        0    SET RULES             0    STOGROUP        0      0      0    UNLOCK REQUEST           38.50      231
PREPARE      1.00        6    SET CURR.PATH         0    SYNONYM         0      0    N/A    QUERY REQUEST             0.00        0
OPEN         1.00        6    SET CURR.PREC.        0    VIEW            0      0    N/A    CHANGE REQUEST            1.00        6
FETCH        2.00       12    CONNECT TYPE 1        0    ALIAS           0      0    N/A    OTHER REQUEST             0.00        0
CLOSE        1.00        6    CONNECT TYPE 2        6    PACKAGE       N/A      0    N/A    LOCK SUSPENSIONS          6.33       38
                              SET CONNECTION        0    PROCEDURE       0      0    N/A    IRLM LATCH SUSPENSIONS    0.50        3
                              RELEASE               0    FUNCTION        0      0    N/A    OTHER SUSPENSIONS         0.00        0
DML-ALL      5.00       30    CALL                  0    TRIGGER         0      0    N/A    TOTAL SUSPENSIONS         6.83       41
                              ASSOC LOCATORS        0    DIST TYPE       0      0    N/A
                              ALLOC CURSOR          0
                              HOLD LOCATOR          0    TOTAL           3      3      0
                              FREE LOCATOR          0    RENAME TBL      0
                              DCL-ALL              12    COMMENT ON      0
                                                         LABEL ON        0
```

Figure 29. One Entry on Long Accounting Report (Part 1 of 2)

```
   LOCATION: STM4D61Y                              DB2 PERFORMANCE MONITOR (V6)                                  PAGE: 1-2
      GROUP: N/P                                    ACCOUNTING REPORT - LONG                        REQUESTED FROM: NOT SPECIFIED
     MEMBER: N/P                                                                                                TO: NOT SPECIFIED
  SUBSYSTEM: Y61Y                                  ORDER: ENDUSER-PRIMAUTH-WSNAME                    INTERVAL FROM: 01/29/99 23:48:01.86
DB2 VERSION: V6                                           SCOPE: MEMBER                                         TO: 01/29/99 23:53:34.20

ENDUSER: HUGO  PRIMAUTH: SOF  WSNAME: WORKSTATNAME

NORMAL TERM.     AVERAGE    TOTAL   ABNORMAL TERM.       TOTAL   IN DOUBT          TOTAL   DRAIN/CLAIM     AVERAGE    TOTAL
---------------  --------  --------  -----------------  --------  --------------  --------  --------------  --------  --------
NEW USER            0.00        0   APPL.PROGR. ABEND        0   APPL.PGM ABEND        0   DRAIN REQUESTS     0.00        0
DEALLOCATION        1.00        6   END OF MEMORY            0   END OF MEMORY         0   DRAIN FAILED       0.00        0
APPL.PROGR. END     0.00        0   RESOL.IN DOUBT           0   END OF TASK           0   CLAIM REQUESTS    53.83      323
RESIGNON            0.00        0   CANCEL FORCE             0   CANCEL FORCE          0   CLAIM FAILED       0.00        0
DBAT INACTIVE       0.00        0
RRS COMMIT          0.00        0

DATA CAPTURE       AVERAGE    TOTAL   DATA SHARING         AVERAGE    TOTAL   QUERY PARALLELISM            AVERAGE    TOTAL
-----------------  --------  --------  ------------------  --------  --------  ----------------------------  --------  --------
IFI CALLS MADE         N/P       N/P   GLOBAL CONT RATE(%)     N/C       N/A   MAXIMUM MEMBERS USED              N/A        1
RECORDS CAPTURED       N/P       N/P   FALSE CONT RATE(%)      N/C       N/A   MAXIMUM DEGREE                    N/A       10
LOG RECORDS READ       N/P       N/P   L-LOCKS XES RATE(%)    0.00         0   GROUPS EXECUTED                  2.00       12
ROWS RETURNED          N/P       N/P   LOCK REQ   - PLOCKS    0.00         0    RAN AS PLANNED                  2.00       12
RECORDS RETURNED       N/P       N/P   UNLOCK REQ - PLOCKS    0.00         0    RAN REDUCED                     0.00        0
DATA DESC. RETURN      N/P       N/P   CHANGE REQ - PLOCKS    0.00         0    ONE DB2-COORDINATOR = NO        0.00        0
TABLES RETURNED        N/P       N/P   LOCK REQ   - XES       0.00         0    ONE DB2-ISOLATION LEVEL         0.00        0
DESCRIBES              N/P       N/P   UNLOCK REQ - XES       0.00         0    SEQUENTIAL-CURSOR               0.00        0
                                       CHANGE REQ - XES       0.00         0    SEQUENTIAL-NO ESA SORT          0.00        0
                                       SUSPENDS   - IRLM      0.00         0    SEQUENTIAL-NO BUFFER            0.00        0
                                       SUSPENDS   - XES       0.00         0    SEQUENTIAL-ENCLAVE SERVICES     0.00        0
                                       SUSPENDS   - FALSE     0.00         0   MEMBER SKIPPED (%)                N/C      N/A
                                       INCOMPATIBLE LOCKS     0.00         0   DISABLED BY RLF                  0.00        0
                                       NOTIFY MSGS SENT       0.00             REFORM PARAL-CONFIG              0.00        0
                                                                               REFORM PARAL-NO BUF              0.00        0

STORED PROCEDURES  AVERAGE    TOTAL   UDF        AVERAGE    TOTAL   TRIGGERS           AVERAGE    TOTAL
-----------------  --------  --------  ---------  --------  --------  -----------------  --------  --------
CALL STATEMENTS       0.00        0   EXECUTED      0.00        0   STATEMENT TRIGGER     0.00        0
ABENDED               0.00        0   ABENDED       0.00        0   ROW TRIGGER           0.00        0
TIMED OUT             0.00        0   TIMED OUT     0.00        0   SQL ERROR OCCUR       0.00        0
REJECTED              0.00        0   REJECTED      0.00        0

LOGGING              AVERAGE    TOTAL   ROWID          AVERAGE    TOTAL   RID LIST             AVERAGE    TOTAL
-------------------  --------  --------  -------------  --------  --------  -------------------  --------  --------
LOG RECORDS WRITTEN     0.00        0   DIRECT ACCESS     0.00        0   USED                    0.00        0
TOT BYTES WRITTEN       0.00        0   INDEX USED        0.00        0   FAIL-NO STORAGE         0.00        0
                                        TS SCAN USED      0.00        0   FAIL-LIMIT EXCEEDED     0.00        0

AVERAGE SU          CLASS 1         CLASS 2   OPTIMIZATION         AVERAGE    TOTAL   MISCELLANEOUS        AVERAGE    TOTAL
------------   --------------  --------------  ------------------  --------  --------  -------------------  --------  --------
CPU                77505.67        77449.67   REOPTIMIZATION         0.00        0   MAX STOR LOB VALUES     0.00        0
 AGENT             18000.33        17944.33   PREP_STMT_MATCH        0.00        0
  NONNESTED        18000.33        17944.33   PREP_STMT_NO_MATCH     0.00        0
  STORED PRC           0.00            0.00   IMPLICIT_PREPARES      0.00        0
  UDF                  0.00            0.00   PREP_FROM_CACHE        0.00        0
  TRIGGER              0.00            0.00   CACHE_LIMIT_EXCEED     0.00        0
 PAR.TASKS         59505.33        59505.33   PREP_STMT_PURGED       0.00        0

BP0                     AVERAGE    TOTAL   BP1                     AVERAGE    TOTAL   TOT4K                   AVERAGE    TOTAL
---------------------  --------  --------  ---------------------  --------  --------  ---------------------  --------  --------
BPOOL HIT RATIO (%)        7.80       N/A   BPOOL HIT RATIO (%)       99.84       N/A   BPOOL HIT RATIO (%)       29.94       N/A
HPOOL HIT RATIO (%)        0.00       N/A   HPOOL HIT RATIO (%)        0.00       N/A   HPOOL HIT RATIO (%)        0.00       N/A
GETPAGES                8014.67     48088   GETPAGES                2538.67     15232   GETPAGES               10553.33     63320
GETPAGES-FAILED            5.00        30   GETPAGES-FAILED            5.00        30   GETPAGES-FAILED            5.00        30
BUFFER UPDATES             0.00         0   BUFFER UPDATES          2585.33     15512   BUFFER UPDATES          2585.33     15512
SYNCHRONOUS WRITE          0.00         0   SYNCHRONOUS WRITE          0.00         0   SYNCHRONOUS WRITE          0.00         0
SYNCHRONOUS READ          84.33       506   SYNCHRONOUS READ           0.00         0   SYNCHRONOUS READ          84.33       506
SEQ. PREFETCH REQS       234.50      1407   SEQ. PREFETCH REQS         5.00        30   SEQ. PREFETCH REQS       239.50      1437
LIST PREFETCH REQS         0.00         0   LIST PREFETCH REQS         0.00         0   LIST PREFETCH REQS         0.00         0
DYN. PREFETCH REQS         1.00         6   DYN. PREFETCH REQS         0.00         0   DYN. PREFETCH REQS         1.00         6
PAGES READ ASYNCHR.     7305.33     43832   PAGES READ ASYNCHR.        4.00        24   PAGES READ ASYNCHR.     7309.33     43856
HPOOL WRITES               0.00         0   HPOOL WRITES               0.00         0   HPOOL WRITES               0.00         0
HPOOL WRITES-FAILED        0.00         0   HPOOL WRITES-FAILED        0.00         0   HPOOL WRITES-FAILED        0.00         0
PAGES READ ASYN-HPOOL      0.00         0   PAGES READ ASYN-HPOOL      0.00         0   PAGES READ ASYN-HPOOL      0.00         0
HPOOL READS                0.00         0   HPOOL READS                0.00         0   HPOOL READS                0.00         0
HPOOL READS-FAILED         0.00         0   HPOOL READS-FAILED         0.00         0   HPOOL READS-FAILED         0.00         0


ACCOUNTING REPORT COMPLETE
```

Figure 29. One Entry on Long Accounting Report (Part 2 of 2)

Explain

This section contains information about how to produce explain reports. Use these reports to see the access path DB2 has chosen for a plan together with the associated catalog information.

The unit of reporting in this report set is an explainable SQL statement. The statement can be identified by:

- *Query number*

  Use this specification when you want to investigate the access path of a particular SQL statement that has been executed.
- *Text of the SQL statement*

  Use this specification when you want to investigate how an SQL statement will execute.
- *Plan or package to which the statement belongs*

  Use this specification when you want to investigate all or selected SQL statements within a plan or package.
- *QMF query name*

  Use this specification when you want to investigate a saved QMF query.

  Note that if the explain plan has been bound with a planname other than the default planname DGOPMEX, you need to specify this plan name using the GLOBAL PLANEXPLAIN option.

The following is an example of the EXPLAIN command:

```
⋮
EXPLAIN
     SQLSTMT
        SELECT *
          FROM (DSN8610.EMP;)
        LEVEL(BASIC) SSID(APC0)
⋮
```

The command requests that a basic explain report (containing row PLAN_TABLE and access path information) be produced for the SQL statement specified in the TEXT option.

The explain report shown in Figure 30 on page 69 is produced.

Product-Sensitive Programming Interface

```
ACTUAL AT:01/15/99 12:25:15      DB2PM (V6)               PAGE       :     1-1
                              EXPLAIN SQL STATEMENT       DB2 RELEASE: V6
LOCATION: SYSDSN5                                         USER AUTHID: PMDEV
SUBSYSTEM:SG51                      BASIC                 CURR.SQLID : PMDEV


 SQL STATEMENT TEXT :



             SELECT * FROM DSN8610.EMP

-------------------------------------------------------------------------------
DSN_STATEMENT_TABLE
EXPLAIN TIME: 1999-01-15-12.25.15.890000
QUERY NO:     N/A , APPL.NAME   :
                    PROGNAME    : DGO@MAIN
                    COLLID      : PMDEVX
                    GROUP MEMBER: SG51
                    STMT TYPE   : SELECT

COST CATEGORY:  A
REASON:
ESTIMATED PROCESSOR COST (MS):             7
          PROCESSOR COST (SU):            13

-------------------------------------------------------------------------------
PLAN TABLE DATA    OWNER: PMDEV
BIND TIME : 1999-01-15-12.25.15.890000

QUERYNO:    N/A , ACC. TYPE:   R, PLAN NO:   1, TABLE NAME: EMP
Q BLOCK NO.:   1, MATCHCOLS:   0, TAB. NO:   1,   -  OWNER: DSN8610
DATE: 1999-01-15, PLAN:         , METHOD :   0, INDEX NAME:
TIME: 12:25:15.8, DBRM: DYN.STMT, IDXONLY : NO,   -  OWNER:
TS LOCKMODE:  IS, COL.FUNC. :   , PREFETCH:  S, 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.: SG51

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
-------------------------------------------------------------------------------

THE ACCESS PATH CHOSEN BY DB2 AT 12:25:15.8   ON 1999-01-15

+------------------------------------------------------------------+
! TABLE SPACE SCAN - NO INDEX WILL BE USED                           !
! STANDARD SEQUENTIAL PREFETCH WILL BE PERFORMED                     !
! LOCK MODE IS SHARE LOCK FOR THE PAGE                               !
! PAGE RANGE SCAN WILL NOT BE USED                                   !
!                                                                    !
!                                                                    !
+------------------------------------------------------------------+
```

Figure 30. Explain Report (Part 1 of 2)

```
REPORT ON:01/15/99 12:25:15        DB2 PM (V6)                SUMMARY PAGE
                                      EXPLAIN
                                   SUMMARY REPORT        USER AUTHID: PMDEV



THE FOLLOWING   1 DB2 PM EXPLAIN REQUESTS WERE PROCESSED:              PAGE NO



  1:    SG51 SQL STMT
             BASIC REPORT REQUESTED

                     SELECT * FROM DSN8610.EMP
                          * TABLE SPACE SCAN-NO INDEX WILL BE USED         1-1



DB2PM EXPLAIN PROCESSING COMPLETED.
```

Figure 30. Explain Report (Part 2 of 2)

End of Product-Sensitive Programming Interface

The volume of output produced by explain can be large, but there are various options to control the amount of detail on the explain reports. The most important ones are LEVEL and PACKLIMIT.

Use the LEVEL option to define how detailed a report you want. You can, for example, produce a summary report that contains one line per SQL statement.

Also, if you have specified plan as the EXPLAIN object, you can use the PACKLIMIT option to limit the number of packages to be explained.

You can specify any number of EXPLAIN commands in the DB2 PM job stream. A separate report is produced for every command.

Source Explain

This section contains information about how to use source explain.

You use DB2 PM source explain to explain SQL statements that are embedded in a source program or SPUFI input. Source explain is performed from within the ISPF/PDF editor. The supported languages are:
- Assembler
- C/370(TM)
- COBOL
- FORTRAN
- PL/I
- SPUFI

Note: Before you activate source explain, make sure that the DB2 load library is allocated to your TSO ISPF session.

To explain an SQL statement while editing a source program (or SPUFI input), you specify the lines you want explained using the ISPF/PDF editor line prefix command **E**, then you type `EXPLAIN` on the command line and press **Enter**. Ensure that source explain has been installed at your site.

The **E** line prefix command can be used as follows:

E To explain a single line, you type `E` in the prefix area of the line to be scanned for SQL statements.

EE To explain a range of lines, you type `EE` in the prefix area of the first and last lines of the range to be scanned for SQL statements.

E[n] To explain a specific number of lines, type `E[`*n*`]` on the first line of the area to scan, where *n* is the number of lines to be scanned for SQL statements.

When you enter the EXPLAIN command, source explain scans the specified range for valid SQL statements. If a range is not specified, the entire source is scanned. Note that an SQL statement is processed even if the SQL statement text exceeds the specified range.

Figure 31 is an example of how to explain an SQL statement while editing COBOL source code. To explain a range of source from line 3040 to 3160, type `EE` in the line prefix area of lines 3040 and 3160 as shown in Figure 31. Type `EXPLAIN` on the command line and press **Enter** to explain the SQL statements within the specified range of lines.

```
EDIT ---- SYS1.DSN610.SDSNSAMP(DSN8BC3) - 01.00 -------------- COLUMNS 001 072
003010       *** CURSOR LISTS ALL EMPLOYEE NAMES WITH A PATTERN (%) OR (_)
003020       *** FOR LAST NAME
003030
EE3040                 EXEC SQL DECLARE TELE2 CURSOR FOR
003050                          SELECT *
003060                          FROM   VPHONE
003070                          WHERE  LASTNAME  LIKE :LNAME-WORK
003080                            AND  FIRSTNAME LIKE :FNAME-WORK
003090                      END-EXEC.
003100
003110       *** CURSOR LISTS ALL EMPLOYEES WITH A SPECIFIC
003120       *** LAST NAME
003130
003140                 EXEC SQL DECLARE TELE3 CURSOR FOR
003150                          SELECT *
EE3160                          FROM   VPHONE
003170                          WHERE  LASTNAME   =   :LNAME
003180                            AND  FIRSTNAME LIKE :FNAME-WORK
003190                       END-EXEC.
003200       /
003210       /*****************************************************
003220       * FIELDS SENT TO MESSAGE ROUTINE                      *
003230       *****************************************************
003240        01  MAJOR                    PIC X(07) VALUE 'DSN8BC3'.
003250
003260        01  MSGCODE                  PIC X(4).
COMMAND ===> explain                                        SCROLL ===> CSR
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 31. Source Explain Example

If the *Always display this window* field has been selected, the Source Explain Options window is displayed. Otherwise, the SQL Statement Selection window is displayed.

Source Explain Options Window

You can display this window from the ISPF/PDF editor in one of the following ways:

- By typing `EXPLAIN OPTIONS` on the command line and pressing **Enter**
- By typing `EXPLAIN` on the command line and pressing **Enter** after the *Always display this window* field is selected
- From the SQL Statement Selection panel by typing `OPTIONS` on the command line and pressing **Enter**

You use the Source Explain Options window (Figure 32 on page 73) to specify the source explain processing options, such as the language of the source code you are editing, the subsystem ID of the explaining DB2, the SQLID to be used, and the degree of parallelism.

```
DGOMYWSO          Source Explain Options

  Update fields as required then press Enter

  Local DB2 Subsystem . . . . . . . . . ____
  Current Server Location . . . . . . . ________________
  Current SQLID . . . . . . . . . . . . ________
  Query number  . . . . . . . . . . . . _________
  _  Set current degree to ANY
  /  Always display this window
  Define source language  . . . . . . . _  1. Assembler
                                           2. C/370
                                           3. COBOL
                                           4. FORTRAN
                                           5. PL/I
                                           6. SPUFI


 F1=Help       F2=Split      F3=Exit       F9=Swap
F12=Cretriev  F13=Help      F14=Split     F15=Exit
F21=Swap      F24=Cancel
```

Figure 32. Source Explain Options Window

The fields shown on the Source Explain Options window are:

Local DB2 Subsystem
: The local DB2 subsystem to which you want to connect.

Current Server Location
: The DB2 subsystem where you want the source SQL statements to be explained. This field makes it possible that you are connected to a DB2 test subsystem, while you execute the explain on the remote production system. If this field is left blank, it defaults to the local DB2 subsystem.

Current SQLID
: A different SQL authorization ID for qualifying the unqualified tables in the SQL statements being explained. A different SQL authorization ID is also used to qualify the plan table to be accessed. If this field is not specified, it defaults to your user ID.

Query number
: A number that identifies the explain statement. If left blank, the value 999735912 is assigned to this field.

Set current degree to ANY
: To specify whether the SQL statement is eligible for query parallelism. If you enter a slash (/) in this field, the current degree of parallelism is set to ANY. Otherwise, the current degree of parallelism is set to 1.

Always display this window
: To control the display of the Source Explain Options window. If you enter a slash (/) in this field, the Source Explain Options window is displayed on each source explain request. If the field is blank, the Source Explain Options window is not displayed on each source explain request.

Define source language
: The language of the source containing the SQL statements. This field cannot be left blank.

After specifying the required information, press **Enter** to proceed to the SQL Statement Selection panel.

SQL Statement Selection Panel

You use this window to view a summarized list of the valid SQL statements within the specified source area. If there are many SQL statements listed, you can use the scrolling keys **F7** (Up) and **F8** (Down) to browse the list.

```
DGOMYWSS                SQL Statement Selection         ROW 1 TO 2 OF 2

This panel summarizes the SQL statements found in the following source
module that can be explained.
                     SYS1.DSN610.SDSNSAMP(DSN8BC3)

Select an SQL statement then press Enter to process.

    Source
    Line No  SQL Statement Text
_    003040  DECLARE TELE2 CURSOR FOR SELECT * FROM VPHONE WHERE LASTNAM
_    003140  DECLARE TELE3 CURSOR FOR SELECT * FROM VPHONE
-- End of List --









Command ===> ____________________________________________________________
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cretriev F13=Help     F14=Split    F15=Exit     F19=Backward F20=Forward
F21=Swap     F24=Cancel
```

Figure 33. SQL Statement Selection Panel

You can select any SQL statement listed on this window for explain processing. The selected SQL statement is then processed and the Online Monitor is accessed, where you can view the explain output on the DB2 Explain Output panel. Refer to the *DB2 PM Online Monitor User's Guide* for more information.

You can use the OPTIONS command to access the Source Explain Options window from the SQL Statement Selection panel, where you can change various source explain processing options. To do this, enter `OPTIONS` on the command line and press **Enter**. This command is only available on the SQL Statement Selection panel.

System Parameters

This section contains information about how to produce system parameters reports. These reports show how the DB2 subsystem you are monitoring is configured.

There is no DB2 PM command used to generate system parameters reports. The reports are generated automatically for each DB2 PM execution if you have specified data set information for the ddname SYSPRMDD in your JCL and the input data is available.

The following figure shows an example of SYSPRMDD definition.

```
⋮
//SYSPRMDD DD  SYSOUT=*
⋮
```

A report entry is produced for every location present in the input data. An entry is also printed if DB2 was restarted with changed system parameters or a change to the system parameters was detected when the statistics interval was reached.

The buffer pool attributes of a DB2 subsystem can be changed while the system is active. These changes are recorded in the system parameters report in the order of occurrence, if the appropriate DB2 trace class has been active. For the DB2 Version 4 data sharing environment, group buffer pool attributes are also reported in the system parameters reports. Note that the group bufferpool attributes (IFCID 230) are available if statistics class S is active.

An example of a system parameters report is shown in Figure 34 on page 76.

```
   LOCATION: GSDLHULB                        DB2 PERFORMANCE MONITOR (V6)                         PAGE: 1-1
      GROUP: DSN2                              SYSTEM PARAMETERS REPORT
     MEMBER: HULB
  SUBSYSTEM: HULB                                                                         ACTUAL FROM: 07/03/98 09:02:55.58
DB2 VERSION: V6

 STORAGE SIZES INSTALLATION PARAMETERS (DSNTIPC,DSNTIPE)
 -------------------------------------------------------
 MAX NO OF USERS CONCURRENTLY RUNNING IN DB2 (CTHREAD)........70
 MAX NO OF TSO CONNECTIONS (IDFORE)...........................40
 MAX NO OF BATCH CONNECTIONS (IDBACK).........................20
 MAX NO OF REMOTE CONNECTIONS (CONDBAT).......................32
 MAX NO OF CONCURRENT REMOTE ACTIVE CONNECTIONS (MAXDBAT).....32
 MAXIMUM SIZE OF EDM POOL IN BYTES (EDMPOOL)...........6,540,288
 MAXIMUM SIZE OF SORT POOL IN BYTES (SRTPOOL)............897,024
 MAXIMUM SIZE OF RID POOL IN BYTES (MAXRBLK)...........4,489,216
 MAX EDM POOL DATA SPACE SIZE in KBYTES..................123,456
 3990 CACHE (SEQCACH).....................................BYPASS
 UTILITY CACHE OPTION (SEQPRES)...............................NO
 MAXIMUM KEPT DYNAMIC STATEMENTS (MAXKEEPD)................5,000
 CONTRACT THREAD STORAGE (CONTSTOR)..........................YES

 TRACING, CHECKPOINT & PSEUDO-CLOSE PARAMETERS (DSNTIPN)
 -----------------------------------------------------------
 START AUDIT TRACE (AUDITST)..................................NO
 START GLOBAL TRACE (TRACSTR).................................16
 TRACE TABLE SIZE IN 4K BYTES (TRACTBL).......................16
 START SMF ACCOUNTING (SMFACCT)................................1
 START SMF STATISTICS (SMFSTAT)..........................1,3,4,5
 STATISTICS TIME INTERVAL IN MINUTES (STATIME).................5
 ONLINE DATASET STATISTICS TIME INTERVAL IN MIN.(DSSTIME)..1.440
 START MONITOR TRACE (MON)....................................NO
 MONITOR BUFFER SIZE IN BYTES (MONSIZE)....................8,192
 CHECKPOINT FREQUENCY (LOGLOAD)...........................50,000
 PSEUDO-CLOSE FREQUENCY (PCLOSEN)..............................5
 PSEUDO-CLOSE TIMER (PCLOSET).................................10
 UR CHECK FREQUENCY (URCHKTH)................................167
 LIMIT BACKOUT (LBACKOUT)...................................AUTO
 BACKOUT DURATION (BACKODUR)..................................15

 OPERATOR FUNCTIONS INSTALLATION PARAMETERS (DSNTIPO)
 ----------------------------------------------------..........1
 WTO ROUTE CODES (ROUTCDE).....................................1
 RESOURCE LIMIT FACILITY AUTOMATIC START (RLF)................NO
 RESOURCE LIMIT SPECIFICATION TABLE SUFFIX (RLFTBL)...........01
 RESOURCE LIMIT SPEC TABLE ERROR ACTION (RLFERR).........NOLIMIT
 AUTO BIND (ABIND)...........................................YES
 ALLOW EXPLAIN AT AUTOBIND (ABEXP)...........................YES
 DPROP SUPPORT (EDPROP)......................................ANY
 SITE TYPE (SITETYP)...................................LOCALSITE
 TRACKER SITE (SPRMTRKR).....................................YES
 READ COPY2 ARCHIVE (ARC2FRST)...............................YES

 DATA INSTALLATION PARAMETERS (DSNTIPA2)
 ---------------------------------------
 ICF CATALOG QUALIFIER (CATALOG)............................DSN2

 IRLM INSTALLATION PARAMETERS (DSNTIPI)
 --------------------------------------
 IRLM SUBSYSTEM NAME (IRLMSID)..............................I513
 IRLM RESOURCE TIMEOUT IN SECONDS (IRLMRWT)...................60
 IRLM AUTOMATIC START (IRLMAUT)..............................YES
 IRLM START PROCEDURE NAME (IRLMPRC)....................HULBIRLM
 SECONDS DB2 WILL WAIT FOR IRLM START (IRLMSWT)..............600
 UTILITY TIMEOUT FACTOR (UTIMOUT)..............................6
 U LOCK FOR RR OR RS (RRULOCK)................................NO
 X LOCK FOR UPDATE/DELETE (SPRMXLUD).........................YES

 ARCHIVE LOG INSTALLATION PARAMETERS (DSNTIPA)
 ---------------------------------------------
 NUMBER OF ARCHIVE LOG COPIES (TWOARCH)........................1
 COPY 1 PREFIX (ARCPFX1)......................DSN2.HULB.ARCHLOG1
 COPY 2 PREFIX (ARCPFX2)......................DSN2.HULB.ARCHLOG2
 CATALOG ARCHIVE DATASETS (CATALOG)..........................YES
 COPY1 ARCHIVE LOG DEVICE TYPE (UNIT).......................DASD
 COPY2 ARCHIVE LOG DEVICE TYPE (UNIT2)...................'BLANK'
 SPACE ALLOCATION METHOD (ALCUNIT)......................CYLINDER
 PRIMARY SPACE ALLOCATION (PRIQTY)............................48
 SECONDARY SPACE ALLOCATION (SECQTY)...........................2
 ARCHIVE LOG BLOCK SIZE IN BYTES (BLKSIZE)................20,480
 MAXIMUM READ TAPE UNITS (MAXRTU)..............................2
 TAPE UNIT DEALLOCATION PERIOD(LOGPDMIN,LOGPDSEC)..............0
 MAX NUMBER OF DATASETS RECORDED IN BSDS (MAXARCH).........1,000
 FIRST ARCHIVE COPY MASS STG GROUP NAME (MSGVP)...........'NONE'
 SECOND ARCHIVE COPY MASS STG GROUP NAME (MSGVP2).........'NONE'
 DAYS TO RETAIN ARCHIVE LOG DATA SETS (ARCRETN)............9,999
 ISSUE WTOR BEFORE MOUNT FOR ARCHIVE VOLUME (ARCWTOR)........YES
 COMPACT DATA (COMPACT).......................................NO
 TIMESTAMP ARCHIVE LOG DATA SETS (TSTAMP).....................NO
 EXTENDED DATESTAMP INDICATOR (TIMESTAMP)....................YES
 QUIESCE PERIOD (QUIESCE)......................................5

 DISTRIBUTED DATA FACILITY PANEL 1 (DSNTIPR)
 -------------------------------------------
 DDF STARTUP OPTION (DDF)................................COMMAND
 RLST ACCESS ERROR (RLFERRD).............................NOLIMIT
 RESYNCHRONIZATION INTERVAL IN MINUTES (RESYNCH)...............2
 DBAT STATUS (CMTSTAT)....................................ACTIVE
 HOP SITE AUTHORIZATION (HOPAUTH).......................PKGOWNER
 IDLE THREAD TIMEOUT INTERVAL (IDTHTOIN).......................0
 EXTENDED SECURITY (EXTSEC)..................................N/P

 DISTRIBUTED DATA FACILITY PANEL 2 (DSNTIP5)
 -------------------------------------------
 TCP/IP ALREADY VERIFIED (TCPALVER)...........................NO
 TCP/IP KEEPALIVE (TCPKPALV)..............................ENABLE
 EXTRA BLOCKS REQ (EXTRAREQ)..................................15
 EXTRA BLOCKS SRV (EXTRASRV)..............................22,617
```

Figure 34. System Parameters Report (Part 1 of 4)

```
   LOCATION: GSDLHULB                          DB2 PERFORMANCE MONITOR (V6)                             PAGE: 1-2
      GROUP: DSN2                                SYSTEM PARAMETERS REPORT
     MEMBER: HULB
  SUBSYSTEM: HULB                                                                             ACTUAL FROM: 07/03/98 09:02:55.58
DB2 VERSION: V6

 PROTECTION INSTALLATION PARAMETERS (DSNTIPP)
 --------------------------------------------
 DB2 AUTHORIZATION ENABLED (AUTH)............................YES
 SYSTEM ADMINISTRATOR 1 AUTHORIZATION ID (SYSADM).........SYSADM
 SYSTEM ADMINISTRATOR 2 AUTHORIZATION ID (SYSADM2)..........SCFA
 SYSTEM OPERATOR 1 AUTHORIZATION ID (SYSOPR1).............SYSOPR
 SYSTEM OPERATOR 2 AUTHORIZATION ID (SYSOPR2)...............SCFA
 DEFAULT (UNKNOWN) USER AUTHORIZATION ID (DEFLTID).......IBMUSER
 RESOURCE LIMIT TABLE CREATOR AUTH ID (RLFAUTH).........SYSIBMAB
 ARCHIVE LOG RACF PROTECTION (PROTECT)........................NO
 PACKAGE AUTHORIZATION CACHE SIZE (CACHEPAC)...............5,000
 PLAN AUTHORIZATION CACHE SIZE (AUTHCACH)..................1,024
 ROUTINE AUTHORIZATION CACHE SIZE (CACHERAC)..............32,768
 BIND NEW PACKAGE (BINDNV)...............................BINDADD

 LOG INSTALLATION PARAMETERS (DSNTIPL,DSNTIPH)
 ---------------------------------------------
 NUMBER OF ACTIVE LOG COPIES (TWOACTV)..........................1
 INPUT BUFFER SIZE IN K BYTES (INBUFF)........................28
 OUTPUT BUFFER SIZE IN K BYTES (OUTBUFF).....................400
 WRITE THRESHOLD ON NO OF FILLED BUFFER (WRTHRSH).............20

 DATA DEFINITION CONTROL SUPPORT (DSNTIPZ)
 -----------------------------------------
 INSTALL DD CONTROL (RGFINSTL)................................NO
 CONTROL ALL APPLICATIONS (RGFDEDPL)..........................NO
 REQUIRE FULL NAMES (RGFFULLQ)...............................YES
 UNREGISTERED DDL DEFAULT (RGFDEFLT)......................ACCEPT
 REGISTER TABLE OWNER (RGFCOLID)........................DSNRGCOL
 DDL REGISTRATION DATABASE NAME (RGFDBNAM)..............DSNRGFDB
 APPL REGISTRATION TABLE NAME (RGFNMPRT).......DSN_REGISTER_APPL
 OBJECT REGISTRATION TABLE NAME (RGFNMORT).....DSN_REGISTER_OBJT
 ESCAPE CHARACTER (RGFESCP)................................X'40'

 ROUTINE PARAMETERS (DSNTIPX)
 ----------------------------
 DB2 PROC NAME (STORPROC)...............................HULBSPAS
 MAX ABEND COUNT (STORMAX)......................................0
 TIMEOUT VALUE (STORTIME).....................................300
 WLM ENVIRONMENT (WLMENV).......................................W

 DATA SHARING PARAMETERS (DSNTIPK,DSNTIPA1)
 ------------------------------------------
 GROUP NAME (GRPNAME)........................................DSN2
 MEMBER NAME (MEMBNAME)......................................HULB
 DATA SHARING ENABLED.........................................YES
 MAXIMUM NUMBER OF MEMBERS....................................248
 PARALLELISM ASSISTANT (ASSIST)...............................YES
 PARALLELISM COORDINATOR (COORDNTR)...........................YES
 SU CONVERSION FACTOR.......................................7,850

 OTHER SYSTEM PARAMETERS
 -----------------------
 MAX NO OF DATA SETS CONCURRENTLY IN USE (DSMAX)...........1,800
 DUAL BSDS MODE (TWOBSDS)......................................YES
 IRLM INITIALIZATION INQUIRY BY DB2............................10
 ASYNC DRAIN START - %DSMAX.....................................1
 ASYNC DRAIN STOP - %DSMAX......................................3
 CHECKPOINTS BETWEEN LEVEL ID UPDATES (DLDFREQ).................0
 IMS/BMP TIMEOUT FACTOR (BMPTOUT)...............................0
 IMS/DLI TIMEOUT FACTOR (DLITOUT)...............................1
 STATIC DESCRIBE (DESCSTAT)...................................YES
 ROLL UP PARALLEL TASK ACCOUNTING (PTASKROL)..................YES

 LOCK ESCALATION PARAMETERS (DSNTIPJ)
 ------------------------------------
 MAX PAGE OR ROW LOCKS PER TABLE SPACE (NUMLKTS)...........1,000
 MAX PAGE OR ROW LOCKS PER USER (NUMLKUS).................10,000
 WAIT FOR RETAINED LOCKS (RETLWAIT)...........................N/P

 APPLICATION PROGRAMMING DEFAULTS PANEL 1 (DSNTIPF)
 --------------------------------------------------...........37
 EBCDIC SBCS CCSID (SCCSID)....................................37
 EBCDIC MBCS CCSID (MCCSID)...................................N/P
 EBCDIC GBCS CCSID (GCCSID)...................................N/P
 ASCII SBCS CCSID (ASCCSID)...................................819
 ASCII MBCS CCSID (AMCCSID)...................................N/P
 ASCII GBCS CCSID (AGCCSID)...................................N/P
 DECIMAL POINT OPTION (DECIMAL)............................PERIOD
 DEFAULT CHARSET (CHARSET)...............................ALPHANUM
 DEFAULT DELIMITER (DELIM).................................APOST
 DEFAULT ENCODING SCHEME (ENSCHEME)........................EBCDIC
 DEFAULT HOST LANGUAGE (DEFLANG)............................COBOL
 DEFAULT MIXED GRAPHIC (MIXED).................................NO
 DEFAULT SQL DELIMITER (SQLDELI)..........................DEFAULT
 DIST SQL STRING DELIMITER (DSQLDELI)......................APOST
 DECIMAL ARITHMETIC (DECARTH)..................................15
 USE FOR DYNAMIC RULES........................................YES

 APPLICATION PROGRAMMING DEFAULTS PANEL 2 (DSNTIP4)
 --------------------------------------------------
 DATE FORMAT (DATE)...........................................ISO
 TIME FORMAT (TIME)...........................................ISO
 LOCAL DATE LENGTH (DATELEN)..................................N/A
 LOCAL TIME LENGTH (TIMELEN)..................................N/A
 STD SQL LANGUAGE (STDSQL).....................................NO
 CURRENT DEGREE (CDSSRDEF)......................................1
 CACHE DYNAMIC SQL (CACHEDYN)..................................NO
 OPTIMIZATION HINTS ALLOWED (OPTHINT)..........................NO
 VARCHAR FROM INDEX (RETVLCFK).................................NO
 RELEASE CURSOR HOLD LOCKS (RELCURHL)..........................NO
```

Figure 34. System Parameters Report (Part 2 of 4)

```
   LOCATION: GSDLHULB                          DB2 PERFORMANCE MONITOR (V6)                             PAGE: 1-3
      GROUP: DSN2                                SYSTEM PARAMETERS REPORT
     MEMBER: HULB
  SUBSYSTEM: HULB                                                                             ACTUAL FROM: 07/03/98 09:02:55.58
DB2 VERSION: V6

 SIZES PANEL 2 (DSNTIP7)
 -----------------------
 USER LOB VALUE STORAGE (LOBVALA)......................2,031,616
 SYSTEM LOB VALUE STORAGE (LOBVALS)....................2,031,616

 DATABASES AND SPACES STARTED AUTOMATICALLY (DSNTIPS)
 ----------------------------------------------------
 ALL
```

Figure 34. System Parameters Report (Part 3 of 4)

```
    LOCATION: GSDLHULB                          DB2 PERFORMANCE MONITOR (V6)                         PAGE: 1-4
      GROUP: DSN2                                SYSTEM PARAMETERS REPORT
     MEMBER: HULB
  SUBSYSTEM: HULB                                                                               ACTUAL FROM: 07/03/98 09:02:55.58
DB2 VERSION: V6

 GROUP BUFFER POOL PARAMETERS
 -----------------------------------------------------------------------------------------------
 TIMESTAMP            07/03/98 09:12:25.51          CURRENT DIRECTORY TO DATA RATIO            5
 MEMBER                                 M2          CLASS CASTOUT THRESHOLD (%)               10
 GBP ID                               GBP0          GBP CASTOUT THRESHOLD (%)                 50
 ALLOCATED GBP SIZE (4K)               256          GBP CHECKPOINT INTERVAL (MIN)              8
 ACTUAL DIRECTORY                      942          GBP CACHE SETTING                        YES
 ACTUAL DATA ENTRY                     187          AUTO REC                                 YES
 PENDING DIRECTORY TO DATA RATIO         5
 MODE                               DUPLEX
 SEC-GBP ALLOC                       12345
 SEC-GBP ALLOC DIRECTORY ENTRY       23451
 SEC GBP DATA ENTRY                  34512

 ALTER GROUP BUFFER POOL COMMAND ISSUED                                                       OLD          NEW
 -----------------------------------------------------------------------------------------------------------
 TIMESTAMP            07/03/98 09:14:31.29          CURRENT DIRECTORY TO DATA RATIO           5            4
 MEMBER                                 M2          CLASS CASTOUT THRESHOLD (%)              10           11
 GBP ID                               GBP0          GBP CASTOUT THRESHOLD (%)                50           51
                                                    GBP CHECKPOINT INTERVAL (MIN)             8            9
                                                    GBP CACHE SETTING                       YES          YES
                                                    AUTO REC                                YES           NO

BUFFER POOL PARAMETERS
------------------------------------------------------------------------------------------------------------
TIMESTAMP             07/03/98 09:12:25.51          VIRTUAL POOL TYPE                                PRIMARY
BP ID                                  BP0          VIRTUAL POOL SIZE                                  3,000
                                                    HIPERPOOL SIZE                                         0
                                                    VIRTUAL POOL SEQUENTIAL THRESHOLD                     80
                                                    HIPERPOOL SEQUENTIAL THRESHOLD                        80
                                                    HORIZONTAL DEFERRED WRITE THRESHOLD                   50
                                                    VERTICAL DEFERRED WRITE THRESHOLD (PERCENTAGE)        10
                                                    VERTICAL DEFERRED WRITE THRESHOLD (BUFFERS)            0
                                                    VIRTUAL POOL PARALLEL SEQUENTIAL THRESHOLD            50
                                                    ASSISTING PARALLEL SEQUENTIAL THRESHOLD                0
                                                    CASTOUT                                               NO
                                                    PAGE STEAL METHOD                                    LRU

ALTER BUFFER POOL COMMAND ISSUED                                                                         OLD          NEW
-------------------------------------------------------------------------------------------------------------------------
TIMESTAMP             07/03/98 09:14:31.29          VIRTUAL POOL TYPE                                PRIMARY       DSPACE
BP ID                                  BP0          VIRTUAL POOL SIZE                                  2,000        3,000
                                                    HIPERPOOL SIZE                                         0            0
                                                    VIRTUAL POOL SEQUENTIAL THRESHOLD                     80           80
                                                    HIPERPOOL SEQUENTIAL THRESHOLD                        80           80
                                                    HORIZONTAL DEFERRED WRITE THRESHOLD                   50           50
                                                    VERTICAL DEFERRED WRITE THRESHOLD (PERCENTAGE)        10           10
                                                    VERTICAL DEFERRED WRITE THRESHOLD (BUFFERS)            0            0
                                                    VIRTUAL POOL PARALLEL SEQUENTIAL THRESHOLD            50           50
                                                    ASSISTING PARALLEL SEQUENTIAL THRESHOLD                0           20
                                                    CASTOUT                                               NO           NO
                                                    PAGE STEAL METHOD                                    LRU         FIFO
```

Figure 34. System Parameters Report (Part 4 of 4)

Statistics

This section contains information on how to produce statistics reports. Use statistics reports to monitor the overall performance of your system.

Note that the information included in the statistics reports can be modified. To find out how, refer to “Tailoring Report Layouts” on page 115.

Short Report

The most commonly used statistics report is the short report. It shows summarized data about the most important statistics categories, including CPU times, SQL, locking, log, EDM pool, and buffer pool activity. If your system has been involved in distributed activity, the report also shows DDF data.

To generate this report, all you need to specify is:

```
⋮
STATISTICS
⋮
```

In this report, data is by default summarized for every location during a statistics collection interval. One entry on the statistics report is always at least two pages long.

```
   LOCATION: DSNDG0G                           DB2 PERFORMANCE MONITOR (V6)                                PAGE: 2-1
      GROUP: DSNDG0G                             STATISTICS REPORT - SHORT                      REQUESTED FROM: NOT SPECIFIED
     MEMBER: DG2G                                                                                           TO: NOT SPECIFIED
  SUBSYSTEM: DG2G                                                                                  INTERVAL FROM: 04/03/99 18:35:00.88
DB2 VERSION: V6                                      SCOPE: MEMBER                                          TO: 04/03/99 18:45:01.12

---- HIGHLIGHTS ----------------------------------------------------------------------------------------------------------------------
INTERVAL START: 04/03/99 18:35:00.88    INTERVAL ELAPSED: 10:00.24469    INCREMENTAL BINDS      :     0.00    DBAT QUEUED:       N/P
INTERVAL END  : 04/03/99 18:45:01.12    OUTAGE ELAPSED  :    0.000000    AUTH SUCC.W/OUT CATALOG: 10554.00    DB2 COMMAND:     38.00
SAMPLING START: 04/03/99 18:35:00.88    TOTAL THREADS   :       37.00    BUFF.UPDT/PAGES WRITTEN:    11.39    TOTAL API  :      0.00
SAMPLING END  : 04/03/99 18:45:01.12    TOTAL COMMITS   :    27038.00    PAGES WRITTEN/WRITE I/O:    25.18    MEMBER     :       N/A

CPU TIMES                                      TCB TIME        SRB TIME      TOTAL TIME    OPEN/CLOSE ACTIVITY  QUANTITY
-------------------------------  ---------------  ---------------  ---------------    -------------------  --------
SYSTEM SERVICES ADDRESS SPACE          0.857983     1:00.400642     1:01.258625    OPEN DATASETS - HWM   1006.00
DATABASE SERVICES ADDRESS SPACE        0.191683     1:24.905898     1:25.097581    OPEN DATASETS         1006.00
IRLM                                   0.000451        2.650958        2.651408    IN USE DATA SETS       137.00
DDF ADDRESS SPACE                           N/P             N/P             N/P

SQL DML   QUANTITY    SQL DCL         QUANTITY    SQL DDL     QUANTITY    LOCKING ACTIVITY  QUANTITY    DATA SHARING LOCKS  QUANTITY
--------  --------    --------------  --------    ----------  --------    ----------------  --------    ------------------  --------
SELECT      131.1K    LOCK TABLE          0.00    CREATES         0.00    DEADLOCKS             0.00    GLB CONT.RATE (%)       0.14
INSERT    76934.00    GRANT               0.00    DROPS           0.00    TIMEOUTS              0.00    FLS CONT.RATE (%)       0.02
UPDATE      100.1K    REVOKE              0.00    ALTERS          0.00    SUSPENSIONS-LOCK     82.00    L-LOCKS RATE (%)        0.05
DELETE     5638.00    SET HOST VAR.       0.00    RENAME TBL      0.00    SUSPENSIONS-OTHR     24.00    LOCK REQ.(P-LOCK)   40764.00
PREPARE       0.00    SET SQLID           0.00    COMMENT ON      0.00    LOCK REQUESTS       204.8K    UNLOCK REQ.(P-LCK)  40746.00
DESCRIBE      0.00    SET DEGREE          0.00    LABEL ON        0.00    UNLOCK REQUEST    77361.00    CHANGE REQ.(P-LCK)      0.00
DESC.TBL      0.00    SET RULES           0.00    TOTAL           0.00    LOCK ESCALAT(SH)      0.00    SYNC.XES - LOCK       179.2K
OPEN        123.3K    SET PATH            0.00                            LOCK ESCALAT(EX)      0.00    SYNC.XES - CHANGE     101.4K
CLOSE     70432.00    SET PRECISION       0.00                            DRAIN REQUESTS        0.00    SYNC.XES - UNLOCK     175.3K
FETCH       250.4K    CONNECT TYPE 1      0.00                            CLAIM REQUESTS      254.9K    ASYN.XES-RESOURCES      0.00
TOTAL       757.9K    CONNECT TYPE 2      0.00                                                          TOTAL SUSPENDS        645.00
                      RELEASE             0.00                                                          P-LCK/NFY ENG.UNAV      0.00
                      SET CONNECTION      0.00                                                          INCOM.RETAINED LCK      0.00
                      ASSOC LOCATORS      0.00                                                          PSET/PART NEGOTIAT      0.00
                      ALLOC CURSOR        0.00                                                          PAGE NEGOTIATION      390.00
                      HOLD LOCATOR        0.00
                      FREE LOCATOR        0.00
                      TOTAL               0.00

RID LIST              QUANTITY    ROW ID          QUANTITY    QUERY PARALLELISM    QUANTITY    PLAN/PACKAGE PROC.   QUANTITY
--------------------  --------    -------------   --------    -------------------  --------    -------------------  --------
MAX BLOCKS ALLOCATED     24.00    DIRECT ACCESS       0.00    MAX DEGREE               0.00    PLAN ALLOC-ATTEMPTS      0.00
CURRENT BLKS ALLOC.       0.00    INDEX USED          0.00    GROUPS EXECUTED          0.00    PLAN ALLOC-SUCCESS       0.00
FAILED-NO STORAGE         0.00    TS SCAN USED        0.00     RAN AS PLANNED          0.00    PACK ALLOC-ATTEMPTS      0.00
FAILED-RDS LIMIT          0.00                                 RAN REDUCED             0.00    PACK ALLOC-SUCCESS       0.00
FAILED-DM LIMIT           0.00                                 FALL TO SEQUENTIAL      0.00    AUTOBIND ATTEMPTS        0.00
FAILED-PROCESS LIMIT      0.00                                 ONE DB2 COORD P=NO      0.00    AUTOBIND SUCCESSFUL      0.00
                                                               ONE DB2 ISO LVL         0.00
                                                              MEMBER SKIPPED (%)        N/C
                                                              REFORM PARAL-CONFIG      0.00
                                                              REFORM PARAL-NO BUF      0.00

STORED PROCEDURES  QUANTITY    UDF        QUANTITY    TRIGGERS             QUANTITY
-----------------  --------    ---------  --------    ------------------   --------
CALL STATEMENTS        0.00    EXECUTED       0.00    STATEMENT TRIGGER       0.00
PROCEDURE ABENDED      0.00    ABENDED        0.00    ROW TRIGGER             0.00
CALL TIMED OUT         0.00    TIMED OUT      0.00    SQL ERROR OCCURRED      0.00
CALL REJECTED          0.00    REJECTED       0.00
```

Figure 35. Short Statistics Report (Part 1 of 5)

```
   LOCATION: DSNDG0G                              DB2 PERFORMANCE MONITOR (V6)                                 PAGE: 2-2
      GROUP: DSNDG0G                                STATISTICS REPORT - SHORT                       REQUESTED FROM: NOT SPECIFIED
     MEMBER: DG2G                                                                                               TO: NOT SPECIFIED
  SUBSYSTEM: DG2G                                                                                    INTERVAL FROM: 04/03/99 18:35:00.88
DB2 VERSION: V6                                          SCOPE: MEMBER                                              TO: 04/03/99 18:45:01.12

---- HIGHLIGHTS ----------------------------------------------------------------------------------------------------------------------
INTERVAL START: 04/03/99 18:35:00.88   INTERVAL ELAPSED: 10:00.24469   INCREMENTAL BINDS      :     0.00   DBAT QUEUED:      N/P
INTERVAL END  : 04/03/99 18:45:01.12   OUTAGE ELAPSED  :    0.000000   AUTH SUCC.W/OUT CATALOG: 10554.00   DB2 COMMAND:    38.00
SAMPLING START: 04/03/99 18:35:00.88   TOTAL THREADS   :       37.00   BUFF.UPDT/PAGES WRITTEN:    11.39   TOTAL API  :     0.00
SAMPLING END  : 04/03/99 18:45:01.12   TOTAL COMMITS   :    27038.00   PAGES WRITTEN/WRITE I/O:    25.18   MEMBER     :      N/A

SUBSYSTEM SERVICES                 QUANTITY   LOG ACTIVITY                      QUANTITY   EDM POOL                    QUANTITY
--------------------------------   --------   --------------------------------  --------   -------------------------  --------
IDENTIFY                               1.00   READS SATISFIED-OUTPUT BUFFER       110.00   PAGES IN EDM POOL           5000.00
CREATE THREAD                         37.00   READS SATISFIED-ACTIVE LOG            0.00   FREE PAGES IN FREE CHAIN    4335.00
SIGNON                             26996.00   READS SATISFIED-ARCHIVE LOG           0.00   FAILS DUE TO POOL FULL         0.00
TERMINATE                             38.00                                                PAGES USED FOR CT            388.00
ROLLBACK                               0.00   READ DELAYED-UNAVAILABLE RESOURCE     0.00   PAGES USED FOR PT              0.00
COMMIT PHASE 1                     26997.00   ARCHIVE LOG READ ALLOCATION           0.00   PAGES USED FOR DBD            51.00
COMMIT PHASE 2                     12595.00   ARCHIVE LOG WRITE ALLOCAT.            0.00   PAGES USED FOR SKCT          175.00
READ ONLY COMMIT                   14406.00   BSDS ACCESS REQUESTS                 56.00   PAGES USED FOR SKPT           51.00
UNITS OF RECOVERY GONE INDOUBT         0.00                                                REQUESTS FOR CT SECTIONS       0.00
UNITS OF RECOVERY INDOUBT RESOLV       0.00   WRITE OUTPUT LOG BUFFERS          25520.00   CT NOT IN EDM POOL             0.00
SYNCHS (SINGLE PHASE COMMIT)          37.00   UNAVAILABLE OUTPUT LOG BUFFER         0.00   REQUESTS FOR PT SECTIONS       0.00
QUEUED AT CREATE THREAD                0.00   OUTPUT LOG BUFFER PAGED IN            0.00   PT NOT IN EDM POOL             0.00
SYSTEM EVENT CHECKPOINT                0.00                                                REQUESTS FOR DBD SECTIONS      0.00
                                              LOG RECORDS CREATED                 420.3K   DBD NOT IN EDM POOL            0.00
                                              LOG DATA CREATED (MB)                75.10   PAGES IN DATASPACE             0.00
                                              LOG WRITE I/O REQUESTS                0.00   FREE PAGES IN DATASPACE        0.00
                                              LOG CI WRITTEN                        0.00   FAILS DUE TO DSP FULL          0.00
                                              LOG CI SERIAL WRITE                   0.00   PREP_STMT_HIT_RATIO             N/C
```

Figure 35. Short Statistics Report (Part 2 of 5)

```
   LOCATION: DSNDG0G                              DB2 PERFORMANCE MONITOR (V6)                                 PAGE: 2-3
      GROUP: DSNDG0G                                STATISTICS REPORT - SHORT                       REQUESTED FROM: NOT SPECIFIED
     MEMBER: DG2G                                                                                                TO: NOT SPECIFIED
  SUBSYSTEM: DG2G                                                                                    INTERVAL FROM: 04/03/99 18:35:00.88
DB2 VERSION: V6                                          SCOPE: MEMBER                                              TO: 04/03/99 18:45:01.12

---- HIGHLIGHTS ----------------------------------------------------------------------------------------------------------------------
INTERVAL START: 04/03/99 18:35:00.88   INTERVAL ELAPSED: 10:00.24469   INCREMENTAL BINDS      :     0.00   DBAT QUEUED:      N/P
INTERVAL END  : 04/03/99 18:45:01.12   OUTAGE ELAPSED  :    0.000000   AUTH SUCC.W/OUT CATALOG: 10554.00   DB2 COMMAND:    38.00
SAMPLING START: 04/03/99 18:35:00.88   TOTAL THREADS   :       37.00   BUFF.UPDT/PAGES WRITTEN:    11.39   TOTAL API  :     0.00
SAMPLING END  : 04/03/99 18:45:01.12   TOTAL COMMITS   :    27038.00   PAGES WRITTEN/WRITE I/O:    25.18   MEMBER     :      N/A

BP0    GENERAL          QUANTITY   BP1    GENERAL          QUANTITY   BP2    GENERAL          QUANTITY   TOT4K  GENERAL          QUANTITY
--------------------    --------   --------------------    --------   --------------------    --------   --------------------    --------
BPOOL HIT RATIO (%)        99.86   BPOOL HIT RATIO (%)       100.00   BPOOL HIT RATIO (%)        17.38   BPOOL HIT RATIO (%)        59.53
HPOOL HIT RATIO (%)        10.00   HPOOL HIT RATIO (%)        10.00   HPOOL HIT RATIO (%)        10.00   HPOOL HIT RATIO (%)        10.00
HPOOL R/W RATIO (%)         9.00   HPOOL R/W RATIO (%)         9.00   HPOOL R/W RATIO (%)         9.00   HPOOL R/W RATIO (%)         9.00
GETPAGES-SEQ&RANDOM     60723.00   GETPAGES-SEQ&RANDOM       545.8K   GETPAGES-SEQ&RANDOM       582.2K   GETPAGES-SEQ&RANDOM      1188.7K
GETPAGES-SEQ.ONLY       22489.00   GETPAGES-SEQ.ONLY           0.00   GETPAGES-SEQ.ONLY           0.00   GETPAGES-SEQ.ONLY       22489.00
SYNC.READ-SEQ&RANDOM        2.00   SYNC.READ-SEQ&RANDOM        0.00   SYNC.READ-SEQ&RANDOM    77801.00   SYNC.READ-SEQ&RANDOM    77803.00
SYNC.READ-SEQ.ONLY          0.00   SYNC.READ-SEQ.ONLY          0.00   SYNC.READ-SEQ.ONLY          0.00   SYNC.READ-SEQ.ONLY          0.00
SEQ.PREFETCH REQ         2217.00   SEQ.PREFETCH REQ            0.00   SEQ.PREFETCH REQ            0.00   SEQ.PREFETCH REQ         2217.00
SEQ.PREFETCH READS          0.00   SEQ.PREFETCH READS          0.00   SEQ.PREFETCH READS          0.00   SEQ.PREFETCH READS          0.00
PAGES READ-SEQ.PREF.        0.00   PAGES READ-SEQ.PREF.        0.00   PAGES READ-SEQ.PREF.        0.00   PAGES READ-SEQ.PREF.        0.00
LST.PREFETCH REQUEST        0.00   LST.PREFETCH REQUEST        0.00   LST.PREFETCH REQUEST        0.00   LST.PREFETCH REQUEST        0.00
LST.PREFETCH READS          0.00   LST.PREFETCH READS          0.00   LST.PREFETCH READS          0.00   LST.PREFETCH READS          0.00
PAGES READ-LST.PREF.        0.00   PAGES READ-LST.PREF.        0.00   PAGES READ-LST.PREF.        0.00   PAGES READ-LST.PREF.        0.00
DYN.PREFETCH REQUEST       29.00   DYN.PREFETCH REQUEST     3055.00   DYN.PREFETCH REQUEST    17909.00   DYN.PREFETCH REQUEST    20993.00
DYN.PREFETCH READS         14.00   DYN.PREFETCH READS          1.00   DYN.PREFETCH READS      15023.00   DYN.PREFETCH READS      15038.00
PAGES READ-DYN.PREF.       85.00   PAGES READ-DYN.PREF.        7.00   PAGES READ-DYN.PREF.      403.2K   PAGES READ-DYN.PREF.      403.3K
BUFFER UPDATES          43852.00   BUFFER UPDATES          18111.00   BUFFER UPDATES              0.00   BUFFER UPDATES          61963.00
SYNCHRONOUS WRITES          0.00   SYNCHRONOUS WRITES          0.00   SYNCHRONOUS WRITES          0.00   SYNCHRONOUS WRITES          0.00
ASYNCHRONOUS WRITES        11.00   ASYNCHRONOUS WRITES       205.00   ASYNCHRONOUS WRITES         0.00   ASYNCHRONOUS WRITES       216.00
DATA SET OPENS              0.00   DATA SET OPENS              0.00   DATA SET OPENS              0.00   DATA SET OPENS              0.00
HDW THRESHOLD               0.00   HDW THRESHOLD               0.00   HDW THRESHOLD               0.00   HDW THRESHOLD               0.00
VDW THRESHOLD               3.00   VDW THRESHOLD               0.00   VDW THRESHOLD               0.00   VDW THRESHOLD               3.00
DM THRESHOLD                0.00   DM THRESHOLD                0.00   DM THRESHOLD                0.00   DM THRESHOLD                0.00
```

Figure 35. Short Statistics Report (Part 3 of 5)

```
   LOCATION: DSNDG0G                                DB2 PERFORMANCE MONITOR (V6)                                  PAGE: 2-4
      GROUP: DSNDG0G                                  STATISTICS REPORT - SHORT                        REQUESTED FROM: NOT SPECIFIED
     MEMBER: DG2G                                                                                                  TO: NOT SPECIFIED
  SUBSYSTEM: DG2G                                                                                       INTERVAL FROM: 04/03/99 18:35:00.88
DB2 VERSION: V6                                          SCOPE: MEMBER                                             TO: 04/03/99 18:45:01.12

---- HIGHLIGHTS ----------------------------------------------------------------------------------------------------------------------
INTERVAL START: 04/03/99 18:35:00.88    INTERVAL ELAPSED: 10:00.24469    INCREMENTAL BINDS      :     0.00   DBAT QUEUED:      N/P
INTERVAL END  : 04/03/99 18:45:01.12    OUTAGE ELAPSED  :    0.000000    AUTH SUCC.W/OUT CATALOG: 10554.00   DB2 COMMAND:    38.00
SAMPLING START: 04/03/99 18:35:00.88    TOTAL THREADS   :       37.00    BUFF.UPDT/PAGES WRITTEN:    11.39   TOTAL API  :     0.00
SAMPLING END  : 04/03/99 18:45:01.12    TOTAL COMMITS   :    27038.00    PAGES WRITTEN/WRITE I/O:    25.18   MEMBER     :      N/A

GROUP BP0                     QUANTITY    GROUP BP1                     QUANTITY    GROUP TOT4K                   QUANTITY
---------------------------  --------    ---------------------------  --------    ---------------------------  --------
GROUP BP HIT RATIO (%)          50.00    GROUP BP HIT RATIO (%)          50.00    GROUP BP HIT RATIO (%)          50.00
SYN.READ(XI)-DATA RETURNED    3050.00    SYN.READ(XI)-DATA RETURNED   10352.00    SYN.READ(XI)-DATA RETURNED   13402.00
SYN.READ(XI)-NO DATA RETURN      2.00    SYN.READ(XI)-NO DATA RETURN      0.00    SYN.READ(XI)-NO DATA RETURN      2.00
SYN.READ(NF)-DATA RETURNED       0.00    SYN.READ(NF)-DATA RETURNED       0.00    SYN.READ(NF)-DATA RETURNED       0.00
SYN.READ(NF)-NO DATA RETURN      0.00    SYN.READ(NF)-NO DATA RETURN      0.00    SYN.READ(NF)-NO DATA RETURN      0.00
CLEAN PAGES SYN.WRTN             0.00    CLEAN PAGES SYN.WRTN             0.00    CLEAN PAGES SYN.WRTN             0.00
CHANGED PGS SYN.WRTN          6092.00    CHANGED PGS SYN.WRTN         18106.00    CHANGED PGS SYN.WRTN         24198.00
CLEAN PAGES ASYN.WRT             0.00    CLEAN PAGES ASYN.WRT             0.00    CLEAN PAGES ASYN.WRT             0.00
CHANGED PGS ASYN.WRT             5.00    CHANGED PGS ASYN.WRT             2.00    CHANGED PGS ASYN.WRT             7.00
REG.PG LIST (RPL) RQ            18.00    REG.PG LIST (RPL) RQ            38.00    REG.PG LIST (RPL) RQ            56.00
CLEAN PGS READ RPL              23.00    CLEAN PGS READ RPL               0.00    CLEAN PGS READ RPL              23.00
CHANGED PGS READ RPL             4.00    CHANGED PGS READ RPL           263.00    CHANGED PGS READ RPL           267.00
PGS READ FRM DASD AFTER RPL     85.00    PGS READ FRM DASD AFTER RPL      7.00    PGS READ FRM DASD AFTER RPL     92.00
ASYN.READ-DATA RETURNED          0.00    ASYN.READ-DATA RETURNED          0.00    ASYN.READ-DATA RETURNED          0.00
PAGES CASTOUT                  120.00    PAGES CASTOUT                 5248.00    PAGES CASTOUT                 5368.00
EXPLICIT X-INVALIDATIONS         0.00    EXPLICIT X-INVALIDATIONS         0.00    EXPLICIT X-INVALIDATIONS         0.00
CASTOUT CLASS THRESH             2.00    CASTOUT CLASS THRESH            63.00    CASTOUT CLASS THRESH            65.00
GROUP BP CAST.THRESH             0.00    GROUP BP CAST.THRESH             0.00    GROUP BP CAST.THRESH             0.00
CASTOUT ENG.UNAVAIL.             0.00    CASTOUT ENG.UNAVAIL.             0.00    CASTOUT ENG.UNAVAIL.             0.00
WRITE ENG.UNAVAIL.               0.00    WRITE ENG.UNAVAIL.               0.00    WRITE ENG.UNAVAIL.               0.00
READ FAILED-NO STOR.             0.00    READ FAILED-NO STOR.             0.00    READ FAILED-NO STOR.             0.00
WRITE FAILED-NO STOR             0.00    WRITE FAILED-NO STOR             0.00    WRITE FAILED-NO STOR             0.00
```

Figure 35. Short Statistics Report (Part 4 of 5)

```
   LOCATION: DSNDG0G                                DB2 PERFORMANCE MONITOR (V6)                                  PAGE: 2-5
      GROUP: DSNDG0G                                  STATISTICS REPORT - SHORT                        REQUESTED FROM: NOT SPECIFIED
     MEMBER: DG2G                                                                                                  TO: NOT SPECIFIED
  SUBSYSTEM: DG2G                                                                                       INTERVAL FROM: 04/03/99 18:35:00.88
DB2 VERSION: V6                                          SCOPE: MEMBER                                             TO: 04/03/99 18:45:01.12

---- HIGHLIGHTS ----------------------------------------------------------------------------------------------------------------------
INTERVAL START: 04/03/99 18:35:00.88    INTERVAL ELAPSED: 10:00.24469    INCREMENTAL BINDS      :     0.00   DBAT QUEUED:      N/P
INTERVAL END  : 04/03/99 18:45:01.12    OUTAGE ELAPSED  :    0.000000    AUTH SUCC.W/OUT CATALOG: 10554.00   DB2 COMMAND:    38.00
SAMPLING START: 04/03/99 18:35:00.88    TOTAL THREADS   :       37.00    BUFF.UPDT/PAGES WRITTEN:    11.39   TOTAL API  :     0.00
SAMPLING END  : 04/03/99 18:45:01.12    TOTAL COMMITS   :    27038.00    PAGES WRITTEN/WRITE I/O:    25.18   MEMBER     :      N/A

                  TRN SENT  CON SENT  CON QUE  SQL SENT  COM SENT  RBK SENT  ROW SENT  MSG SENT  BYT SENT   LIM BLK  BLK SENT
 REMOTE LOCATION  TRN RECV  CON RECV  SYS BIND SQL RECV  COM RECV  RBK RECV  ROW RECV  MSG RECV  BYT RECV   MSG BUF  BLK RECV
----------------  --------  --------  -------- --------  --------  --------  --------  --------  --------  --------  --------
                   1401.00   1401.00     0.00   4892.00   1401.00      0.00      0.00   7696.00   1260.3K      0.00      0.00
DRDA REMOTE LOCS      0.00      0.00     0.00      0.00      0.00      0.00   6291.00   6295.00    706.1K   1955.00    250.00

STATISTICS REPORT COMPLETE
```

Figure 35. Short Statistics Report (Part 5 of 5)

Exception Report

An efficient way to identify potential problems in your system performance is to use exception processing. The exception reports only show report entries with fields containing values outside limits you have specified. For more information about setting the limits, refer to “Chapter 8. Exception Reporting” on page 95.

To generate the exception report, specify:

```
⋮

STATISTICS
     REPORT
        EXCEPTION
⋮
```

The exception report only contains entries with fields containing values exceeding the limits you have specified. A block of exception information is printed after the report entries.

```
⋮
⋮
***********************************************************************************************************************************
*  TYPE           FIELD ID   FIELD DESCRIPTION                                     BY                      VALUE   THRESHOLD     *
*                            FIELD QUALIFIER                                                                                     *
*  PROBLEM        QBSTRIO    SYNCHRONOUS READS                                     TOTAL                      53 > 50            *
*                            BP0                                                                                                 *
*  WARNING        QBSTRIO    SYNCHRONOUS READS                                     TOTAL                       9 > 1             *
*                            BP2                                                                                                 *
*  WARNING        QBSTRIO    SYNCHRONOUS READS                                     TOTAL                      14 > 1             *
*                            BP7                                                                                                 *
***********************************************************************************************************************************
⋮
⋮
```

Figure 36. Statistics Exception Block

SQL Activity

This section contains information about how to produce SQL Activity traces and reports. SQL Activity traces and reports are used to further investigate problems discovered in accounting or to examine the SQL statements in a program.

For further information about SQL Activity traces and reports as well as a detailed description of the available options, refer to the *DB2 PM Report Reference Volume 2*.

Every Occurrence of an SQL Statement

The SQL Activity trace summarized by occurrence is the most straightforward of the SQL traces.

To generate an SQL Activity trace summarized by occurrence, all you need to specify is:

```
⋮
SQLACTIVITY
⋮
```

For further information on how to produce specific SQL Activity traces and reports, refer to the *DB2 PM Report Reference Volume 2*.

```
   LOCATION: DSNAPC3                            DB2 PERFORMANCE MONITOR (V6)                              PAGE: 1-1
      GROUP: GROUP_1                                 SQL ACTIVITY - TRACE                        REQUESTED FROM: NOT SPECIFIED
     MEMBER: MEMBER_1                                                                                        TO: NOT SPECIFIED
  SUBSYSTEM: APC3                                                                                   ACTUAL FROM: 04/08/99 03:28:52.13
DB2 VERSION: V6
                                                     SUMMARIZED BY OCCURRENCE

             PRIMAUTH: XXASP09           CONNECT : BATCH              CORRNAME: XXASP09F   CONNTYPE: TSO
             ORIGAUTH: XXASP09           PLANNAME: LOCCURHL           CORRNMBR: 'BLANK'    THRDTYPE: ALLIED
             ENDUSER : 1234567890123456  WSNAME  : 123456789012345678 TRANSACT: 12345678901234567890123456789012

TRACE # 1.1                 DB2 LUWID: APCNET.SYDAPC3.X'A44FECDC9ED4'                           ACE ADDRESS: X'02684398'

START TIME: 04/08/99 03:28:52.13  START ELAPSED:               0.028475    START REASON: CREATE THREAD
STOP TIME : 04/08/99 03:29:02.63  STOP ELAPSED :               0.009044    STOP REASON : TERMINATE THREAD

NL     EVENT          TIMESTAMP   ELAP.TIME  TCB  TIME                                     DETAIL
-------------------- ----------- ----------- --------- --------------------------------------------------------------------------------
CALL                 03:28:40.08    4.325677           STMT#   313 PROCEDURE: CSF3SP01                 SQLSTATE: N/P   SQLCODE:     0
                                                       SCHEDULE TIME:    2.279645  SCHEDULE TCB: N/P
PACKAGE                                                M05EC003.PRODCOLL.CSF3SP01.X'15C14A6204FE6D86'
                                                       ACQUIRE(USE)       REOPT(N)  RELEASE(COMMIT)      ISO(CS)  DYNAMICRULES(BIND)
                                                       PREPARE(NODEFER)   KEEPDYNAMIC(NO)    PROTOCOL(NOT_SPEC)    OPTHINT( )
SELECT               03:28:42.35    0.510555           STMT#   109                             ISO(CS) SQLSTATE: 02000 SQLCODE:   100
                                                       REOPTIMIZED(NO)  KEEP UPDATE LOCKS:
                                                       TEXT: PARM CHAR(5) INOUT, CHAR(5) INOUT, CHAR(20) INOUT, DECIMAL(10,3) INOUT,
                                                             SMALLINT INOUT, INTEGER INOUT, CHAR(5) INOUT, VARCHAR(50) INOUT, FLOAT
                                                             INOUT, FLOAT INOUT, VARCHAR(20) INOUT, INTEGER INOUT, CHAR(20) INOUT,
                                                             CHAR(74) INOUT
PREPARE              03:28:42.87    0.445684           STMT#   161                                     SQLSTATE: 00000 SQLCODE:     0
                                                       TEXT: INSERT INTO CATE3TB5_VF VALUES( ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
                                                             ?)
INSERT               03:28:43.31    1.085400           STMT#   171                             ISO(CS) SQLSTATE: 00000 SQLCODE:     0
                                                       REOPTIMIZED(NO)  KEEP UPDATE LOCKS: YES
SELECT               03:28:44.40    0.000861           STMT#   199                             ISO(CS) SQLSTATE: 00000 SQLCODE:     0
                                                       REOPTIMIZED(NO)  KEEP UPDATE LOCKS: YES
DBRM                                                   CSF3AP01
DESCRIBE             03:28:44.40                       STMT#   313                                     SQLSTATE: 00000 SQLCODE:     0
SYNC.                03:28:44.40    0.155893
DESCRIBE             03:28:44.56                       STMT#   512                                     SQLSTATE: 00000 SQLCODE:     0
 1  CALL             09:00:00.00 25:00.00000 25:00.000 STMT#    64 PROCEDURE: MANFREDS STOPROC01       SQLSTATE: N/P   SQLCODE:     0
                                                       SCHEDULE TIME: N/P          SCHEDULE TCB: N/P         SCHEMA : MANF__01
 2  CALL             09:05:00.00 15:00.00000 15:00.000 STMT#   264 PROCEDURE: MANFREDS STOPROC02       SQLSTATE: N/P   SQLCODE:     0
                                                       SCHEDULE TIME: N/P          SCHEDULE TCB: N/P         SCHEMA : MANF__02
 3  CALL             09:10:00.00 5:00.000000 5:00.0000 STMT#   364 PROCEDURE: MANFREDS STOPROC03       SQLSTATE: N/P   SQLCODE:     0
                                                       SCHEDULE TIME: N/P          SCHEDULE TCB: N/P         SCHEMA : MANF__03
 4  INVOKE           09:11:00.00 1:00.000000 1:00.0000 STMT#   464 FUNCTION : MANFREDS USERFUNCT       SQLSTATE: N/P   SQLCODE:     0
                                                       SCHEDULE TIME: N/P          SCHEDULE TCB: N/P         SCHEMA : MANF__04
PACKAGE                                                DB01LOC.NULLID.SQLL16N1.X'4741427143444B4D'
                                                       ACQUIRE(USE)    REOPT(N)  RELEASE(COMMIT)      ISO(CS)  DYNAMICRULES(RUN)
                                                       PREPARE(NODEFER)   KEEPDYNAMIC(NO)    PROTOCOL(NOT_SPEC) OPTHINT(12345678)
SELECT               03:28:44.59    0.004778           STMT#   109                             ISO(CS) SQLSTATE: 02000 SQLCODE:   100
                                                       REOPTIMIZED(NO)  KEEP UPDATE LOCKS: N/P

TRIGGER              03:28:50.01    0.123456       N/A STMT#   180 SCHEMA : TRSCHEMA NAME: TRIGNAME    SQLSTATE: 12345 SQLCODE: 12345
                                                                   ACT    : BEFORE   OPER: UPDATE      GRAN    : STMT  COND    : FALSE

OPEN                 03:28:52.15    0.000208  0.000207 STMT#   189 CURSOR: C1HOLD                  ISO(CS) SQLSTATE: 00000 SQLCODE:     0
                                                       REOPTIMIZED(YES)                                KEEP UPDATE LOCKS: YES

FETCH                03:28:52.18    0.252314  0.002603 STMT#   196 CURSOR: C1HOLD                          SQLSTATE: 00000 SQLCODE:     0

UPDATE               03:28:55.46    0.932496  0.003920 STMT#   213 CURSOR: C1HOLD                  ISO(CS) SQLSTATE: 00000 SQLCODE:     0

INSERT               03:29:01.43    1.137912  0.007616 STMT#   245                        REOPT(N) ISO(CS) SQLSTATE: 00000 SQLCODE:     0

CLOSE                03:29:02.60    0.000451  0.000451 STMT#   256 CURSOR: C1HOLD                          SQLSTATE: 00000 SQLCODE:     0
```

Figure 37. SQL Activity Trace Summarized by Occurrence

The trace shows the SQL statements in the order they were executed for the first ten threads present in the input data. A new page is started when the thread being reported changes. The trace also shows the program or DBRM under which the statements were executed.

Sorting the Data

Use SORTBY to sequence your SQL events according to the criterion that is most useful to you. Note that the sort order is always descending.

You can specify one entry of SORTBY for each TRACE subcommand. One of the following options can be specified for SORTBY:

- ELAPSEDTIME — Elapsed time
- TCBTIME — TCB time
- SCANS — Number of scans
- ROWSPROC — Rows processed
- PAGESCAN — Pages scanned
- SORTRECS — Records sorted
- SORTWORK — Workfiles sorted
- SORTPASS — Merge passes for each sort
- IOREQS — I/O requests
- IOTIME — Elapsed time for each I/O request
- SUSP — Lock suspensions
- SUSPTIME — Elapsed time for each lock suspension
- EXITS — Number of exits
- EXITTIME — Elapsed time for each exit

To sort your SQL activity trace by TCB time, specify:

```
⋮
SQLACTIVITY
     TRACE
        SORTBY (TCBTIME)
⋮
```

```
   LOCATION: DSNAPC3                         DB2 PERFORMANCE MONITOR (V6)                          PAGE: 1-1
      GROUP: GROUP_1                              SQL ACTIVITY - TRACE                    REQUESTED FROM: NOT SPECIFIED
     MEMBER: MEMBER_1                                                                                TO: NOT SPECIFIED
  SUBSYSTEM: APC3                                                                           ACTUAL FROM: 04/08/99 03:28:52.13
DB2 VERSION: V6
                                           SUMMARIZED BY OCCURRENCE, SORTED BY TCBTIME

             PRIMAUTH: XXASP09           CONNECT : BATCH              CORRNAME: XXASP09F   CONNTYPE: TSO
             ORIGAUTH: XXASP09           PLANNAME: LOCCURHL           CORRNMBR: 'BLANK'    THRDTYPE: ALLIED
             ENDUSER : 1234567890123456  WSNAME  : 123456789012345678 TRANSACT: 12345678901234567890123456789012

TRACE # 1.1                 DB2 LUWID: APCNET.SYDAPC3.X'A44FECDC9ED4'                          ACE ADDRESS: X'02684398'

START TIME: 04/08/99 03:28:52.13  START ELAPSED:               0.028475    START REASON: CREATE THREAD
STOP TIME : 04/08/99 03:29:02.63  STOP ELAPSED :               0.009044    STOP REASON : TERMINATE THREAD

       EVENT          TIMESTAMP   ELAP.TIME  TCB  TIME                                      DETAIL
-------------------- ----------- ----------- --------- -----------------------------------------------------------------------------

TRIGGER              03:28:51.35    0.000215 10.045763 STMT#   180 SCHEMA: TRSCHEMA NAME: TRIGNAME    SQLSTATE: 12345 SQLCODE: 12345
                                                                    ACT   : BEFORE   OPER: UPDATE      GRAN    : STMT  COND    : FALSE

UPDATE               03:28:55.46    0.932496  0.003920 STMT#   213 CURSOR: C1HOLD            ISO(CS) SQLSTATE: 00000 SQLCODE:     0
                                                       DBRM: LOCCURHL

FETCH                03:28:52.18    0.252314  0.002603 STMT#   196 CURSOR: C1HOLD                    SQLSTATE: 00000 SQLCODE:     0
                                                       DBRM: LOCCURHL

UPDATE               03:28:56.69    0.000675  0.000676 STMT#   228 CURSOR: C1HOLD   REOPT(Y) ISO(CS) SQLSTATE: 00000 SQLCODE:     0
                                                       DBRM: LOCCURHL

FETCH                03:29:02.57    0.000518  0.000507 STMT#   251 CURSOR: C1HOLD                    SQLSTATE: 01501 SQLCODE:     0
                                                       DBRM: LOCCURHL

CLOSE                03:29:02.60    0.000451  0.000451 STMT#   256 CURSOR: C1HOLD                    SQLSTATE: 00000 SQLCODE:     0
                                                       DBRM: LOCCURHL

FETCH                03:28:56.69    0.000387  0.000387 STMT#   225 CURSOR: C1HOLD                    SQLSTATE: 00000 SQLCODE:     0
                                                       DBRM: LOCCURHL

FETCH                03:28:55.45    0.000349  0.000351 STMT#   204 CURSOR: C1HOLD                    SQLSTATE: 00000 SQLCODE:     0
                                                       DBRM: LOCCURHL

OPEN                 03:28:52.15    0.000208  0.000207 STMT#   189 CURSOR: C1HOLD            ISO(CS) SQLSTATE: 00000 SQLCODE:     0
                                                       DBRM: LOCCURHL
```

Figure 38. SQL Activity Trace Summarized by Occurrence Sorted by TCB Time

The SQL activity data is summarized by each occurrence by default. The data is presented in descending sequence of TCB time, and the program or DBRM name is embedded in every statement.

Summarizing Further

You might not be interested in every single occurrence of an SQL statement. Use the SUMMARIZEBY option to collect your SQL data, for example, by program name.

You can specify one entry of SUMMARIZEBY for each TRACE subcommand. The following events can be specified:
- STMTNO (statement number)
- CURSOR (cursor)
- PROGRAM (program)
- STMTTYPE (statement type)
- ALL (all of the above).

To produce an SQL activity trace summarized by cursor and sorted by average TCB time, specify:

```
⋮

SQLACTIVITY
     TRACE
        SUMMARIZEBY (CURSOR)
        SORTBY(TCBTIME)
⋮
```

```
   LOCATION: DSNAPC3                          DB2 PERFORMANCE MONITOR (V6)                          PAGE: 1-1
      GROUP: GROUP_1                               SQL ACTIVITY - TRACE                    REQUESTED FROM: NOT SPECIFIED
     MEMBER: MEMBER_1                                                                                 TO: NOT SPECIFIED
  SUBSYSTEM: APC3                                                                            ACTUAL FROM: 04/08/99 03:28:52.13
DB2 VERSION: V6
                                            SUMMARIZED BY CURSOR, SORTED BY AVERAGE TCBTIME

             PRIMAUTH: XXASP09           CONNECT : BATCH              CORRNAME: XXASP09F   CONNTYPE: TSO
             ORIGAUTH: XXASP09           PLANNAME: LOCCURHL           CORRNMBR: 'BLANK'    THRDTYPE: ALLIED
             ENDUSER : 1234567890123456  WSNAME  : 123456789012345678  TRANSACT: 12345678901234567890123456789012

TRACE # 1.1                 DB2 LUWID: APCNET.SYDAPC3.X'A44FECDC9ED4'                        ACE ADDRESS: X'02684398'

START TIME: 04/08/99 03:28:52.13  START ELAPSED:               0.028475    START REASON: CREATE THREAD
STOP TIME : 04/08/99 03:29:02.63  STOP ELAPSED :               0.009044    STOP REASON : TERMINATE THREAD

       EVENT            COUNT     TOT.ELAPS  TOTAL TCB                                   DETAIL
                                  AET/EVENT  TCB/EVENT
-------------------- ----------- ----------- --------- ---------------------------------------------------------------------------

C2NOHOLD                        3    7.853205  0.029523 DBRM: LOCCURHL
                                     2.617735  0.009841 STMTTYPE             COUNT   AET/OCCUR TCB/OCCUR              COMMITS:     4
                                                        UPDATE                   1    0.230461  0.004741
                                                        OPEN                     3    0.001238  0.000251
                                                        FETCH                    3    1.001623  0.002264
                                                        DELETE                   1    4.614161  0.017236

C1HOLD                          1    1.187397  0.009101 DBRM: LOCCURHL
                                     1.187397  0.009101 STMTTYPE             COUNT   AET/OCCUR TCB/OCCUR              COMMITS:     4
                                                        UPDATE                   2    0.466585  0.002298
                                                        OPEN                     1    0.000208  0.000207
                                                        FETCH                    4    0.063392  0.000962
                                                        CLOSE                    1    0.000451  0.000451

#   245                         1    1.137912  0.007616 INSERT                        REOPT(N) ISO(CS)
                                     1.137912  0.007616 DBRM: LOCCURHL
```

Figure 39. SQL Activity Trace Summarized by Cursor and Average TCB Time

The trace shows the totals for different SQL statement types summarized for each cursor belonging to the thread. The information is sorted alphabetically by program or DBRM name and by cursor names (where present) within the DBRM.

Workload Details

If you are interested in non-SQL information associated with the execution of SQL statements, you can request workload details.

Workload details are available on all summary levels. The workload figures are applied to the event being summarized. The following workload details can be requested for each event:

- HILITE (workload highlights)
- SCAN (scan, RID list, and query parallelism activity)
- SORT (sort activity)
- IO (I/O activity)
- LOCK (lock suspension and page and row locking activity)
- EXIT (exit activity)
- DCAP (data capture activity)
- ACCT (accounting)
- ALL (all workload activity)
- NONE (no workload activity).

Any combination of parameters can be requested including ALL or NONE. NONE is the default.

The following information is supplied by default, when present and workload is requested:
- MINIBIND (minibind activity
- UDF (user-defined functions)

If you specify WORKLOAD(ALL), as shown in the following example, you will get an SQL activity trace that lists all available workload information.

```
⋮
SQLACTIVITY
     TRACE
        SUMMARIZEBY (CURSOR)
        SORTBY(TCBTIME)
        WORKLOAD(ALL)
⋮
```

```
   LOCATION: DSNAPC3                          DB2 PERFORMANCE MONITOR (V6)                               PAGE: 1-1
      GROUP: GROUP_1                               SQL ACTIVITY - TRACE                       REQUESTED FROM: NOT SPECIFIED
     MEMBER: MEMBER_1                                                                                    TO: NOT SPECIFIED
  SUBSYSTEM: APC3                                                                            ACTUAL FROM: 04/08/99 03:28:52.13
DB2 VERSION: V6
                                   SUMMARIZED BY CURSOR, SORTED BY AVERAGE TCBTIME, WITH ALL WORKLOAD

             PRIMAUTH: XXASP09           CONNECT : BATCH              CORRNAME: XXASP09F   CONNTYPE: TSO
             ORIGAUTH: XXASP09           PLANNAME: LOCCURHL           CORRNMBR: 'BLANK'    THRDTYPE: ALLIED
             ENDUSER : 1234567890123456  WSNAME  : 123456789012345678 TRANSACT: 12345678901234567890123456789012

TRACE # 1.1                 DB2 LUWID: APCNET.SYDAPC3.X'A44FECDC9ED4'                        ACE ADDRESS: X'02684398'

START TIME: 04/08/99 03:28:52.13  START ELAPSED:               0.028475    START REASON: CREATE THREAD
STOP TIME : 04/08/99 03:29:02.63  STOP ELAPSED :               0.009044    STOP REASON : TERMINATE THREAD

       EVENT            COUNT     TOT.ELAPS  TOTAL TCB                                    DETAIL
                                  AET/EVENT  TCB/EVENT
-------------------- ----------- ----------- --------- ----------------------------------------------------------------------------

C2NOHOLD                       3    7.853205  0.029523 DBRM: LOCCURHL
                                    2.617735  0.009841 STMTTYPE            COUNT   AET/OCCUR TCB/OCCUR              COMMITS:     4
                                                       UPDATE                  1    0.230461  0.004741
                                                       OPEN                    3    0.001238  0.000251
                                                       FETCH                   3    1.001623  0.002264
                                                       DELETE                  1    4.614161  0.017236
      --- WORKLOAD HILITE -----------------------------------------------------------------------------------------------------
      SCANS   :     8  RECS/SORT:  3.00  I/O REQS:        1  SUSPENDS   :        2  EXITS    :        2  AMS               :        1
      ROWSPROC:     8  WORK/SORT:  2.00  AET/I/O : 1.374752  AET/SUSP   : 0.485483  AET/EXIT : 0.048234  AET/AMS           : 0.094745
      PAGESCAN:    47  PASS/SORT:  2.00  DATACAPT:      YES  RIDS UNUSED:        2  CHECKCON : REJECTED  DEGREE REDUCTION :        3
      LOB_PAGSCAN:    12345          LOB_UPD_PAGE:    12345
      --- SCAN ACTIVITY -------------------------------------------------------------------------------------------------------
                                 ------ROWS------  --QUALIFIED AT--  --------- -ROWS----------- --MASS- --PAGES- --------RI-------
      DATABASE  PAGESET    SCANS  PROCESS  EXAMINE  STAGE 1  STAGE 2  INSERTS   UPDATES  DELETES DELETES  SCANNED    SCANS  DELETES
      MEMBER    TYPE
      DSNDB01   SCT02          6        6        6        6        0        0         0        0       0       12        0        0
      SE12      INDX
      DSNDB01   SCT02          6        6        6        0        0        0         0        0       0        4        0        0
      SE12      SEQD
      DSNDB06   SYSPLAN        1        1        1        1        0        0         0        0       0        2        0        0
      SE12      INDX
      DSNDB06   SYSPLAN        1        1        1        0        0        0         0        0       0        1        0        0
      SE12      SEQD
      NHDBASE1  NHTSPAC1       1        4        4        4        4        0         2        0       0        5        0        0
      SE12      SEQD
      TOTAL                   15       18       18       11        4        0         2        0       0       24        0        0
```

Figure 40. SQL Activity Trace Summarized by Cursor with All Workload Sorted by Average TCB TIME (Part 1 of 4)

```
   LOCATION: DSNAPC3                            DB2 PERFORMANCE MONITOR (V6)                               PAGE: 1-2
      GROUP: GROUP_1                                 SQL ACTIVITY - TRACE                       REQUESTED FROM: NOT SPECIFIED
     MEMBER: MEMBER_1                                                                                      TO: NOT SPECIFIED
  SUBSYSTEM: APC3                                                                                 ACTUAL FROM: 04/08/99 03:28:52.13
DB2 VERSION: V6
                                   SUMMARIZED BY CURSOR, SORTED BY AVERAGE TCBTIME, WITH ALL WORKLOAD

             PRIMAUTH: XXASP09          CONNECT : BATCH              CORRNAME: XXASP09F   CONNTYPE: TSO
             ORIGAUTH: XXASP09          PLANNAME: LOCCURHL           CORRNMBR: 'BLANK'    THRDTYPE: ALLIED
             ENDUSER : 1234567890123456  WSNAME  : 123456789012345678  TRANSACT: 12345678901234567890123456789012

TRACE # 1.1                 DB2 LUWID: APCNET.SYDAPC3.X'A44FECDC9ED4'                        ACE ADDRESS: X'02684398'

START TIME: 04/08/99 03:28:52.13  START ELAPSED:               0.028475    START REASON: CREATE THREAD
STOP TIME : 04/08/99 03:29:02.63  STOP ELAPSED :               0.009044    STOP REASON : TERMINATE THREAD

       EVENT           COUNT     TOT.ELAPS  TOTAL TCB                                    DETAIL
                                 AET/EVENT  TCB/EVENT
-------------------- ----------- ----------- --------- ----------------------------------------------------------------------------
      --- MINIBIND -------------------------------------------------------------------------------------------------------
      QUERYNO  : 1383      PLANNAME      : DSNTEP61           COST          : 35        PARALLELISM_DISABLED : N/A
      QBLOCKNO : 2         COLLID        : DSNTEP61           PROGNAME      : DSNTEP61  CONSISTENCY_TOKEN    : 15769AE806DB8B8E
      APPLNAME : N/P       WHEN_OPTIMIZE : 'BLANK'            OPT_HINT_IDENT: N/P       OPTIMIZE_HINTS_USED  : YES
      UNITS    : 12345     MILLI_SEC     : 12345              COST_CATEGORY : N/P
      BIND_TIME: 04/08/99 03:28:55.211328  VERSION: N/P
      .........................................................................................................................
      PLANNO          : 1                   METHOD    : FIRST TABLE ACCESSED  SORTN_UNIQ      : NO     SORTC_UNIQ     : NO
      DATABASE        : DSNDB04             NEXTSTEP  : NOT APPLICABLE        SORTN_JOIN      : NO     SORTC_JOIN     : NO
      OBJECT          : 21                  ACCESSTYPE: TABLE SPACE SCAN (R)  SORTN_ORDERBY   : NO     SORTC_ORDERBY  : NO
      CREATOR         : X                   PAGE_RANGE      : NO              SORTN_GROUPBY   : NO     SORTC_GROUPBY  : NO
      TNAME           : TBUF0401            JOIN_TYPE       : NO              SORTN_PGROUP_ID : 0      SORTC_PGROUP_ID: 0
      CORRELATION_NAME: N/P                 MERGE_JOIN_COLS : 0               ACCESS_DEGREE   : 0      JOIN_DEGREE    : 0
      TSLOCKMODE      : IS                  PARALLELISM_MODE: NO              ACCESS_PGROUP_ID: 0      JOIN_PGROUP_ID : 0
      ACCESS_NAME     : N/A                 ACCESS_CREATOR  : N/A             MATCHCOLS       : N/A    PREFETCH       : SEQ
      OPERATION       : N/A                 PREFETCH_INDEX  : N/A             MIXOPSEQ        : N/A    DIRECT_ROW_ACC : N/A
      INDEXONLY       : N/A                 COLUMN_FN_EVAL  : N/A             PAGES_FOR_TABLE : 12345  TAB_CARDINALITY: 123456789A
      --- FUNCTION RESOLUTION(S) -----------------------------------------------------------------------------------------
      QUERYNO  : 1383       PLANNAME : DSNTEP61  COLLECTION_ID : DSNTEP61
      APPLNAME : xxxxxxxx   PROGNAME : xxxxxxxx  CONSIS_TOKEN  : xxxxxxxxxxxxxxxx
      BIND_TIME: 04/08/99 03:28:55.21  VERSION : xxxxxxxx10xxxxxxxx20xxxxxxxx30xxxxxxxx40xxxxxxxx50xxxxxxxxxxxx64
      CURRENT_PATH : ........10........20........30........40........50........60........70........80........90.......100
                     .......110.......120.......130.......140.......150.......160.......170.......180.......190.......200
                     .......210.......220.......230.......240...........254
      ....................................................................................................................
      FUNCT_SCHEMA : xxxxxxxx  FUNCT_NAME : xxxxxxxxxxxxxxxxxx  SPECIFIC_NAME : xxxxxxxxxxxxxxxxxx  FUNCT_TYPE : xxxxx
      VIEW_CREATOR : NAME-111  VIEW_NAME  : xxxxxxxxxxxxxxxxxx  QUERY_BLOCKNO : 53
      FUNCT_TEXT   : ........10........20........30........40........50........60........70........80........90.......100
                     .......110.......120.......130.......140.......150.......160.......170.......180.......190.......200
                     .......210.......220.......230.......240...........254
      --- RID LIST PROCESSING -----------------------------------------------------------------------------------------------
      RIDS IN FINAL LIST:        38  RID LIST USED:     2  UNUSED (LIMIT EXEEDED):     5  UNUSED (NO STORAGE):     1
      DATABASE  PAGESET    THRESHOLD RIDS OBTAINED        RIDS EXCEEDED LIMIT
      NHDBASE1  NHINDEX1        4075            36                          3
      NHDBASE2  NHINDEX2       36000            87                          2
                 AVERAGE    20037.50         61.50                       2.50
      --- I/O ACTIVITY ------------------------------------------------------------------------------------------------------
      DATABASE  PAGESET  - I/O REQUEST -  ------- READ REQUEST  (WITH OR WITHOUT I/O) -------  ------------ WRITE REQUEST ----------
      MEMBER    BP       TOTAL     AET    TOTAL  TYPE   AET/WITH    %WITH PAGE/WITH  %WITHOUT  TOTAL  TYPE  CAST      AET  PAGE/WRIT
      DBPARALL  TSPARALL
      SE12      BP4          3  0.1296        3  SYNCH  0.129597   100.00      1.00     0.00
      WRKSE12   DSN4K01
      SE12      BP0        102  0.0164      102  SYNCH  0.016358   100.00      1.00     0.00
      --- PAGE & ROW LOCKING ---------------------------------------------------------------------------
                                            LOCK   MAX LOCKS   # LOCK   HIGHEST    TS     LOCK AVOID
      MEMBER    DATABASE   PAGESET   COUNT   SIZE              ESCAL      LOCK    TYPE   SUCCESSFUL
      SE12      DSNDB06    SYSDBASE      2  TABLE          0       0        IS   SIMPL          N/P
      SE12      DSNDB06    SYSDBAUT      2  TABLE          0       0        IS   PARTI          N/P
      SE12      DSNDB06    SYSUSER       2    LOB          0       0         1     LOB          N/P
      SE12      DSNDDF     SYSDDF        3  TABLE          0       0              SEG          YES
      SUMMARY : MAX PAGE/ROW/LOB LOCKS HELD 12345   LOCK ESCALATIONS : SHARED 12345   EXCLUSIVE    12345

      TOTAL                         9                        0
      --- EXITS ----------------------------------------------------
      MEMBER     VALIDATION  TOTAL   AET/EXIT   EDIT TOTAL  AET/EXIT
      SE12                       1    N/C                0  0.000060
```

Figure 41. SQL Activity Trace Summarized by Cursor, Sorted by Average TCB Time, with All Workload (Part 2 of 4)

```
   LOCATION: DSNAPC3                                DB2 PERFORMANCE MONITOR (V6)                                      PAGE: 1-3
      GROUP: GROUP_1                                     SQL ACTIVITY - TRACE                               REQUESTED FROM: NOT SPECIFIED
     MEMBER: MEMBER_1                                                                                                   TO: NOT SPECIFIED
  SUBSYSTEM: APC3                                                                                              ACTUAL FROM: 04/08/99 03:28:52.13
DB2 VERSION: V6
                                   SUMMARIZED BY CURSOR, SORTED BY AVERAGE TCBTIME, WITH ALL WORKLOAD

             PRIMAUTH: XXASP09           CONNECT : BATCH               CORRNAME: XXASP09F   CONNTYPE: TSO
             ORIGAUTH: XXASP09           PLANNAME: LOCCURHL            CORRNMBR: 'BLANK'    THRDTYPE: ALLIED
             ENDUSER : 1234567890123456  WSNAME  : 123456789012345678  TRANSACT: 12345678901234567890123456789012

TRACE # 1.1                 DB2 LUWID: APCNET.SYDAPC3.X'A44FECDC9ED4'                         ACE ADDRESS: X'02684398'

START TIME: 04/08/99 03:28:52.13  START ELAPSED:               0.028475    START REASON: CREATE THREAD
STOP TIME : 04/08/99 03:29:02.63  STOP ELAPSED :               0.009044    STOP REASON : TERMINATE THREAD

       EVENT            COUNT     TOT.ELAPS  TOTAL TCB                                 DETAIL
                                  AET/EVENT  TCB/EVENT
-------------------- ----------- ----------- --------- ---------------------------------------------------------------------------

C1HOLD                        1    1.187397  0.009101 DBRM: LOCCURHL
                                   1.187397  0.009101 STMTTYPE            COUNT   AET/OCCUR TCB/OCCUR              COMMITS:     4
                                                      UPDATE                  2    0.466585  0.002298
                                                      OPEN                    1    0.000208  0.000207
                                                      FETCH                   4    0.063392  0.000962
                                                      CLOSE                   1    0.000451  0.000451
      --- WORKLOAD HILITE ---------------------------------------------------------------------------------------------------------
      SCANS   :     8  RECS/SORT:  3.00  I/O REQS:        1  SUSPENDS   :        2  EXITS    :        2  AMS                :      1
      ROWSPROC:     8  WORK/SORT:  2.00  AET/I/O : 1.374752  AET/SUSP   : 0.485483  AET/EXIT : 0.048234  AET/AMS            : 0.094745
      PAGESCAN:    47  PASS/SORT:  2.00  DATACAPT:      YES  RIDS UNUSED:        2  CHECKCON : REJECTED  DEGREE REDUCTION :        3
      LOB_PAGSCAN:    12345          LOB_UPD_PAGE:    12345
      --- SCAN ACTIVITY -----------------------------------------------------------------------------------------------------------
                                   ------ROWS------  --QUALIFIED AT--  --------- -ROWS----------- --MASS- --PAGES- --------RI-------
      DATABASE  PAGESET    SCANS   PROCESS  EXAMINE  STAGE 1  STAGE 2  INSERTS   UPDATES  DELETES DELETES  SCANNED     SCANS  DELETES
      MEMBER    TYPE
      DSNDB01   SCT02          6         6        6        6        0        0         0        0       0       12         0        0
      SE12      INDX
      DSNDB01   SCT02          6         6        6        0        0        0         0        0       0        4         0        0
      SE12      SEQD
      DSNDB06   SYSPLAN        1         1        1        1        0        0         0        0       0        2         0        0
      SE12      INDX
      DSNDB06   SYSPLAN        1         1        1        0        0        0         0        0       0        1         0        0
      SE12      SEQD
      NHDBASE1  NHTSPAC1       1         4        4        4        4        0         2        0       0        5         0        0
      SE12      SEQD
      TOTAL                   15        18       18       11        4        0         2        0       0       24         0        0
      --- SORT ACTIVITY -----------------------------------------------------------------------------------------
      MEMBER      :      SE12  WORKFILES            :    72.00  RECORDS     :      2.45  MAX REQUESTED   :       5
      TOTAL SORTS :         4  INITIAL WORKFILES    :     1.00  RECORD SIZE :  18292.50  AVG REQUESTED   :    3.35
      SORT KEYS   :      2.00  WORKFILES PARTITIONED :    2.00  KEY SIZE    :     30.27  MAX NOT ACQUIRED:       1
      SORT COLUMNS:     25.00  PARTITIONING         :      YES  DATA SIZE   :     92.13  AVG NOT ACQUIRED:    1.00
      AET/SORT    :  2.990676  PARTITIONING & SORTING:     YES  ROWS DELETED:      0.00  MAX RETURN CODE :       4
      SORT TYPE   :   ESA-TAG  PARTITION TYPE       : LASTPASS  MERGE PASSES:      1.00
      --- I/O ACTIVITY ------------------------------------------------------------------------------------------------------------
      DATABASE  PAGESET  - I/O REQUEST -  ------- READ REQUEST  (WITH OR WITHOUT I/O) -------  ------------ WRITE REQUEST ----------
      MEMBER    BP       TOTAL     AET     TOTAL  TYPE   AET/WITH    %WITH PAGE/WITH  %WITHOUT  TOTAL  TYPE  CAST      AET  PAGE/WRIT
      NHDBASE1  NHTSPAC1     3  0.0210         2  SYNCH  0.018509   100.00      1.00      0.00      1 SYNCH  YES  0.027834        1.0
      SE12      BP0                            1  SEQPF  0.026131   100.00
      NHDBASE1  NHTSPAC2     5  0.0378         3  SYNCH  0.053423   100.00      1.00      0.00
      SE12      BP0                            2  SEQPF  0.014461    50.00      4.50     50.00
```

Figure 42. SQL Activity Trace Summarized by Cursor, Sorted by Average TCB Time, with All Workload (Part 3 of 4)

```
   LOCATION: DSNAPC3                             DB2 PERFORMANCE MONITOR (V6)                             PAGE: 1-4
      GROUP: GROUP_1                                  SQL ACTIVITY - TRACE                      REQUESTED FROM: NOT SPECIFIED
     MEMBER: MEMBER_1                                                                                       TO: NOT SPECIFIED
  SUBSYSTEM: APC3                                                                                  ACTUAL FROM: 04/08/99 03:28:52.13
DB2 VERSION: V6
                                   SUMMARIZED BY CURSOR, SORTED BY AVERAGE TCBTIME, WITH ALL WORKLOAD

             PRIMAUTH: XXASP09          CONNECT : BATCH              CORRNAME: XXASP09F   CONNTYPE: TSO
             ORIGAUTH: XXASP09          PLANNAME: LOCCURHL           CORRNMBR: 'BLANK'    THRDTYPE: ALLIED
             ENDUSER : 1234567890123456  WSNAME  : 123456789012345678  TRANSACT: 12345678901234567890123456789012

TRACE # 1.1                 DB2 LUWID: APCNET.SYDAPC3.X'A44FECDC9ED4'                        ACE ADDRESS: X'02684398'

START TIME: 04/08/99 03:28:52.13  START ELAPSED:               0.028475    START REASON: CREATE THREAD
STOP TIME : 04/08/99 03:29:02.63  STOP ELAPSED :               0.009044    STOP REASON : TERMINATE THREAD

       EVENT            COUNT     TOT.ELAPS  TOTAL TCB                                   DETAIL
                                  AET/EVENT  TCB/EVENT
-------------------- ----------- ----------- --------- -------------------------------------------------------------------------------
      --- LOCK SUSPENSION ACTIVITY --------------------------------------------------------------------------------------------------
                                                 ------------- SUSPEND -------------  NORMAL RESUME  TIMEOUT RESUME  DEADLOCK RESUME
      RESOURCE NAME          TYPE      REQUEST  LOCAL LATCH IRLMQ GROUP NOTIFY OTHER  COUNT     AET  COUNT      AET  COUNT       AET
      MEMBER
      NHDBASE1     NHTSPAC1  DATAPAGE  CHANGE       1     0     0     0     0     0      0 13.8728      0      N/C      0       N/C
      SE12
      NHDBASE1     NHTSPAC2  DATAPAGE  LOCK         0     1     0     0     0     0      0 1.98182      0      N/C      0       N/C
      SE12
      NHDBASE1     XDEPT1    OPENLOCK  UNLOCK       1     0     0     0     0     0      0 4.98875      0      N/C      0       N/C
      SE12
      --- PAGE & ROW LOCKING ----------------------------------------------------------------------------
                                                 LOCK   MAX LOCKS   # LOCK   HIGHEST    TS      LOCK AVOID
      MEMBER     DATABASE    PAGESET   COUNT   SIZE                 ESCAL      LOCK    TYPE    SUCCESSFUL
      SE12       DSNDB06     SYSDBASE      2  TABLE          0         0        IS   SIMPL          N/P
      SE12       DSNDB06     SYSDBAUT      2  TABLE          0         0        IS   PARTI          N/P
      SE12       DSNDB06     SYSUSER       2    LOB          0         0         1     LOB          N/P
      SE12       DSNDDF      SYSDDF        3  TABLE          0         0               SEG          YES
      SUMMARY : MAX PAGE/ROW/LOB LOCKS HELD 12345   LOCK ESCALATIONS : SHARED 12345   EXCLUSIVE    12345

      TOTAL                             9                            0

#   245                      1    1.137912  0.007616 INSERT                                 ISO N/A
                                  1.137912  0.007616 DBRM: LOCCURHL
      --- WORKLOAD HILITE -----------------------------------------------------------------------------------------------------------
      SCANS    :     8  RECS/SORT:  3.00  I/O REQS:        1  SUSPENDS    :        2  EXITS     :        2  AMS              :        1
      ROWSPROC:      8  WORK/SORT:  2.00  AET/I/O : 1.374752  AET/SUSP    : 0.485483  AET/EXIT : 0.048234  AET/AMS          : 0.094745
      PAGESCAN:     47  PASS/SORT:  2.00  DATACAPT:      YES  RIDS UNUSED:         2  CHECKCON : REJECTED  DEGREE REDUCTION :        3
      LOB_PAGSCAN:    12345         LOB_UPD_PAGE:    12345
      --- SCAN ACTIVITY -------------------------------------------------------------------------------------------------------------
                             ------ROWS------  --QUALIFIED AT--  --------- -ROWS----------- --MASS- --PAGES- --------RI-------
      DATABASE  PAGESET    SCANS   PROCESS  EXAMINE  STAGE 1  STAGE 2  INSERTS   UPDATES  DELETES DELETES  SCANNED    SCANS  DELETES
      MEMBER    TYPE
      DSNDB01   SCT02          6         6        6        6        0        0         0        0       0       12        0        0
      SE12      INDX
      DSNDB01   SCT02          6         6        6        0        0        0         0        0       0        4        0        0
      SE12      SEQD
      DSNDB06   SYSPLAN        1         1        1        1        0        0         0        0       0        2        0        0
      SE12      INDX
      DSNDB06   SYSPLAN        1         1        1        0        0        0         0        0       0        1        0        0
      SE12      SEQD
      NHDBASE1  NHTSPAC1       1         4        4        4        4        0         2        0       0        5        0        0
      SE12      SEQD
      TOTAL                   15        18       18       11        4        0         2        0       0       24        0        0

      --- I/O ACTIVITY --------------------------------------------------------------------------------------------------------------
      DATABASE  PAGESET  - I/O REQUEST -  ------- READ REQUEST  (WITH OR WITHOUT I/O) -------  ------------ WRITE REQUEST ----------
      MEMBER    BP       TOTAL      AET    TOTAL  TYPE   AET/WITH    %WITH PAGE/WITH  %WITHOUT  TOTAL  TYPE  CAST      AET  PAGE/WRIT
      NHDBASE1  NHTSPAC1     3  0.0210         2  SYNCH  0.018509   100.00      1.00     0.00       1 SYNCH  YES  0.027834        1.0
      SE12      BP0                            1  SEQPF  0.026131   100.00
      NHDBASE1  NHTSPAC2     5  0.0378         3  SYNCH  0.053423   100.00      1.00     0.00
      SE12      BP0                            2  SEQPF  0.014461    50.00      4.50    50.00
      --- EXITS ----------------------------------------------------
      MEMBER     VALIDATION  TOTAL    AET/EXIT    EDIT TOTAL  AET/EXIT
      xxxxxxxx                   1    N/C                  0  0.000060
```

Figure 43. SQL Activity Trace Summarized by Cursor, Sorted by Average TCB Time, with All Workload (Part 4 of 4)

The trace shows all the available workload detail for each SQL event.

Locking

This section contains information about how to produce locking reports. These reports should be produced when you have discovered a locking-related problem in the accounting, statistics, or SQL activity reports.

Lock Suspension Report

If you discover that there are a large number of suspensions in the accounting reports for an application, you can generate a suspension report to analyze the number and type of lock suspensions.

Consider carefully how you want the report ordered, for example, if you want to summarize data for the pageset, database, plan name, authorization ID.

To generate a suspension report ordered by page set within authorization ID, all you need to specify is:

```
⋮
LOCKING
      REPORT
         ORDER (PRIMAUTH-PAGESET)
⋮
```

Ordering by PRIMAUTH gives a breakdown of locking activity for every user, and PAGESET breaks the information down further, providing information on all the locks suspended for a pageset.

The generated report shows the type of lock requests and the type and name of the resource that was locked, as well as the reason for the suspension and how it ended (normal, timeout, or deadlock).

```
   LOCATION: DSNCAT                            DB2 PERFORMANCE MONITOR (V6)                           PAGE: 1-1
      GROUP: DSNCAT                              LOCKING REPORT - SUSPENSION                 REQUESTED FROM: NOT SPECIFIED
     MEMBER: V61B                                                                                        TO: NOT SPECIFIED
  SUBSYSTEM: V61B                                 ORDER: PRIMAUTH-PAGESET                     INTERVAL FROM: 03/05/99 10:00:00.00
DB2 VERSION: V6                                        SCOPE: MEMBER                                     TO: 03/05/99 10:30:00.00

                                                              --SUSPEND REASONS-- ---------- R E S U M E   R E A S O N S -----------
PRIMAUTH           --- L O C K   R E S O U R C E --- TOTAL    LOCAL  GLOB.  S.NFY ---- NORMAL ----  TIMEOUT/CANCEL   --- DEADLOCK ---
 PAGESET           TYPE      NAME                    SUSPENDS LATCH  IRMLQ  OTHER NMBR        AET NMBR         AET NMBR         AET
------------------ --------- ----------------------- -------- -----  -----  ----- ---- ----------- ---- ----------- ---- -----------
SYSOPR
 'BLANK'           X'FF'     DB  =TPCCE1                    1     0      0      0    1    0.021173    0         N/C    0         N/C
                             OB  =TCUST000                        0      0      1
T3270A
 'BLANK'           N/P       N/P                            3     0      0      0    3    0.005693    0         N/C    0         N/C
                                                                  0      0      3
 42                PAGEPLCK  DB  =260                       2     0      0      0    2    8.008926    0         N/C    0         N/C
                             PAGE=X'000017'                       0      0      2
                             BPID=BP2
 44                PAGEPLCK  DB  =260                       2     0      0      0    2    8.366533    0         N/C    0         N/C
                             PAGE=X'00003D'                       0      0      2
                             BPID=BP2
 46                PAGEPLCK  DB  =260                       2     0      0      0    2    4.153638    0         N/C    0         N/C
                             PAGE=X'00001E'                       0      0      2
                             BPID=BP2

*TOTAL*
T3270A                                                      9     0      0      0    9    4.563919    0         N/C    0         N/C
                                                                  0      0      9

*GRAND TOTAL*                                              10     0      0      0   10    4.109645    0         N/C    0         N/C
                                                                  0      0     10

LOCKING REPORT COMPLETE
```

Figure 44. Lock Suspension Report

Lockout Trace

If you want to identify all the participants and resources involved in timeout and deadlock situations, generate a lockout trace.

To produce a lockout trace, specify:

```
:
LOCKING
     TRACE
        LEVEL (LOCKOUT)
:
```

The trace contains an entry for each occurrence of a timeout or deadlock during a specified period of time.

In case of a timeout, the generated trace shows the resource the threads were contending for, as well as detailed information about the thread that was made the timeout victim and the threads that obtained a lock on the resource.

In case of a deadlock, the generated trace contains an entry for each occurrence of a deadlock during a specified period of time. The trace provides details of the resources involved in the deadlock followed by a box containing information about the threads that hold the resource or are waiting to use the resource.

```
   LOCATION: USIBMSYSTDB2                    DB2 PERFORMANCE MONITOR (V6)                          PAGE: 1-1
      GROUP: N/P                                LOCKING TRACE - LOCKOUT                  REQUESTED FROM: NOT SPECIFIED
     MEMBER: N/P                                                                                    TO: NOT SPECIFIED
  SUBSYSTEM: DB2P                                                                          ACTUAL FROM: 04/14/99 21:15:30.53
DB2 VERSION: V6                                   SCOPE: MEMBER                              PAGE DATE: 04/14/99

PRIMAUTH CORRNAME CONNTYPE
ORIGAUTH CORRNMBR INSTANCE     EVENT TIMESTAMP            --- L O C K   R E S O U R C E ---
PLANNAME CONNECT               RELATED TIMESTAMP EVENT    TYPE      NAME                     EVENT SPECIFIC DATA
------------------------------ ----------------- -------- --------- ------------------------ ----------------------------------------
SYSOPR   DBCMD    'BLANK'      21:15:30.53780703 TIMEOUT  DATAPAGE  DB  =DSNDB06             REQUEST =LOCK       UNCONDITIONAL
SYSOPR   0206     AA381CD784F0 N/P                                  OB  =SYSDBASE            STATE   =S          ZPARM INTERVAL=   60
'BLANK'  SSTR                                                       PAGE=X'00000167'         DURATION=MANUAL     INTERV.COUNTER=    1
                                                                                             HASH    =X'0003670C'
                                                                                             ------------ HOLDERS/WAITERS -----------
                                                                                             HOLDER
                                                                                             LUW=USIBMSY.SY10BDB2.AA381CD784F0
                                                                                             MEMBER  =N/P        CONNECT =BATCH
                                                                                             PLANNAME=ABTEPR1    CORRNAME=RUNPL1
                                                                                             DURATION=COMMIT     CORRNMBR='BLANK'
                                                                                             STATE   =X          PRIMAUTH=USER1
                                                                                             ENDUSER =WORK_USER1
                                                                                             WSNAME  =WORKSTATION_NAME1
                                                                                             TRANSACT=VERY LONG TRANSACTION NAME

SYSOPR   DBCMD    'BLANK'      21:18:30.18709909 TIMEOUT  DATABASE  DB  =ABTE2DB2            REQUEST =LOCK       UNCONDITIONAL
SYSOPR   0206     AA381D8398A1 N/P                                                           STATE   =S          ZPARM INTERVAL=   60
'BLANK'  SSTR                                                                                DURATION=ALLOCATN   INTERV.COUNTER=    1
                                                                                             HASH    =X'000024A0'
                                                                                             ------------ HOLDERS/WAITERS -----------
                                                                                             HOLDER
                                                                                             LUW=USIBMSY.SY10BDB2.AA381D8398A1
                                                                                             MEMBER  =N/P        CONNECT =BATCH
                                                                                             PLANNAME=ABTEPR1    CORRNAME=RUNPL1
                                                                                             DURATION=COMMIT     CORRNMBR='BLANK'
                                                                                             STATE   =X          PRIMAUTH=USER2

USER3    DBCMD    'BLANK'      21:30:04.23462747 DEADLOCK                                    COUNTER =    2      WAITERS =    2
USER3    D206     AA382025EFC2 N/P                                                           TSTAMP  =04/13/99 21:30:03.33
BCT      SSTR                                             DATAPAGE  DB  =DSNDB06             HASH    =X'0003670C'
                                                                    OB  =SYSDBASE            ------ BLOCKER IS HOLDER ------*VICTIM*-
                                                                    PAGE=X'00000167'         LUW=USIBMSY.SY10BDB2.AA382025EFC2
                                                                                             MEMBER  =N/P        CONNECT =SSTR
                                                                                             PLANNAME=BCT        CORRNAME=DBCMD
                                                                                             DURATION=MANUAL     CORRNMBR=D206
                                                                                             STATE   =S          PRIMAUTH=USER3
                                                                                             ----------- WAITER ---------------------
                                                                                             LUW=USIBMSY.SY30BDB2.A74AEB7051CA
                                                                                             MEMBER  =N/P        CONNECT =BATCH
                                                                                             PLANNAME=ABTEPR1    CORRNAME=RUNPL1
                                                                                             DURATION=COMMIT     CORRNMBR='BLANK'
                                                                                             REQUEST =CHANGE     PRIMAUTH=USER4
                                                                                             STATE   =X          WORTH   =   18

                                                          DATABASE  DB  =ABTE2DB1            HASH    =X'00002480'
                                                                                             ------ BLOCKER is HOLDER ---------------
                                                                                             LUW=USIBMSY.SY30BDB2.A74AEB7051CA
                                                                                             MEMBER  =N/P        CONNECT =BATCH
                                                                                             PLANNAME=ABTEPR1    CORRNAME=RUNPL1
                                                                                             DURATION=COMMIT     CORRNMBR='BLANK'
                                                                                             STATE   =X          PRIMAUTH=USER4
                                                                                             ----------- WAITER ------------*VICTIM*-
                                                                                             LUW=USIBMSY.SY10BDB2.AA382025EFC2
                                                                                             MEMBER  =N/P        CONNECT =SSTR
                                                                                             PLANNAME=BCT        CORRNAME=DBCMD
                                                                                             DURATION=ALLOCATN   CORRNMBR=D206
                                                                                             REQUEST =LOCK       PRIMAUTH=USER3
                                                                                             STATE   =S          WORTH   =   17
```

Figure 45. Lockout Trace Example

IFCID Frequency Distribution Log and Job Summary Log

To see what IFCIDs were present in the input data for a DB2 PM job and what traces were active during the sampling period, you can produce an IFCID frequency distribution log and a job summary log.

Both logs are generated when you specify a ddname for JOBSUMDD in your job stream.

The IFCID frequency distribution log provides counts of the trace records accumulated by IFCID. There is an *INPUT COUNT* for the number of valid records. There is also a *PROCESSED COUNT* for the number of records that are actually processed after the filters specified in GLOBAL are applied and duplicate records are dropped.

```
   LOCATION: DSNDB0G                            DB2 PERFORMANCE MONITOR (V6)                                  PAGE: 1
      GROUP: DSNDB0G                          IFCID FREQUENCY DISTRIBUTION LOG                       RUN DATE: 04/15/99 08:21:11.18
     MEMBER: DB1G
  SUBSYSTEM: DB1G                                                                                 ACTUAL FROM: 03/19/99 22:34:20.59
DB2 VERSION: V6                                                                                            TO: 03/19/99 22:56:57.67

             INPUT        INPUT      PROCESSED     PROCESSED                       INPUT        INPUT      PROCESSED     PROCESSED
   IFCID     COUNT     PCT OF TOTAL    COUNT     PCT OF TOTAL          IFCID       COUNT     PCT OF TOTAL    COUNT     PCT OF TOTAL
   -------  ----------  ------------  ----------  ------------        -------    ----------  ------------  ----------  ------------
      1            9      9.67%              0      0.00%                 5            10      10.75%             10      17.85%
      2            9      9.67%              0      0.00%               105             9       9.67%              9      16.07%
      3           27     29.03%             27     48.21%               106            10      10.75%              0       0.00%
      4           10     10.75%             10     17.85%               202             9       9.67%              0       0.00%

      TOTAL INPUT TRACE RECORDS  =              93
      TOTAL PROCESSED TRACE RECORDS =           56
```

Figure 46. IFCID Frequency Distribution Log

The job summary log provides a summary of events during DB2 PM execution and other information that helps you interpret the reports.

```
                                               DB2 PERFORMANCE MONITOR (V6)                                  PAGE: 1
                                                      JOB SUMMARY LOG                                RUN DATE: 04/15/99 08:21:03.61


MSG.ID.    LOCATION          GROUP     SSID  MEMBER    TIMESTAMP
           DESCRIPTION
---------  ------------------------------------------------------------------------------------------------------------------------

DGOC4060I  DSNDB0G           DSNDB0G   DB1G  DB1G      03/19/99 22:34:20.59
           DB2 START TRACE NUMBER 01 DB2 SUBSYSTEM ID = DB1G
           TEXT = -START TRACE (STAT )CLASS (* )RMID (* )PLAN (* )AUTHID (* )IFCID (* )BUFSIZE (* )

DGOC4065I  DSNDB0G           DSNDB0G   DB1G  DB1G      03/19/99 22:35:39.68
           DB2 STOP TRACE NUMBER 01 DB2 SUBSYSTEM ID = DB1G
           TEXT = -STOP TRACE (*) TNO (01) COMMENT('TRACE STOPPED BY MODIFY COMMAND')

DGOC4060I  DSNDB0G           DSNDB0G   DB1G  DB1G      03/19/99 22:35:39.68
           DB2 START TRACE NUMBER 01 DB2 SUBSYSTEM ID = DB1G
           TEXT = -MODIFY TRACE (STAT )CLASS (* )TNO (1 )IFCID (* )

DGOC4060I  DSNDB0G           DSNDB0G   DB2G  DB2G      03/19/99 22:34:21.64
           DB2 START TRACE NUMBER 01 DB2 SUBSYSTEM ID = DB2G
           TEXT = -START TRACE (STAT )CLASS (* )RMID (* )PLAN (* )AUTHID (* )IFCID (* )BUFSIZE (* )

DGOC4060I  DSNDB0G           DSNDB0G   DB2G  DB2G      03/19/99 22:34:22.15
           DB2 START TRACE NUMBER 02 DB2 SUBSYSTEM ID = DB2G
           TEXT = -START TRACE (ACCTG )CLASS (1 2 3 )RMID (* )PLAN (* )AUTHID (* )IFCID (* )BUFSIZE (* )


DGOC9200I  ACCOUNTING REDUCE COMPLETED. SUMMARY OF REDUCED DATA FOLLOWS
           LOCATION          GROUP     SSID  MEMBER    INTERVAL START       INTERVAL END            COUNT
           ----------------  --------  ----  --------  -------------------  --------------------  --------
           DSNDB0G           DSNDB0G   DB1G  DB1G      03/19/99 22:35:07.26  03/19/99 22:56:45.89        27
           DSNDB0G           DSNDB0G   DB2G  DB2G      03/19/99 22:47:05.09  03/19/99 22:47:05.09         1

DGOC4005I  DSNDB0G           DSNDB0G   DB1G  DB1G
           NUMBER OF RECORDS PROCESSED WITHOUT A CPU HEADER WAS 56

DGOC4010I  DSNDB0G           DSNDB0G   DB1G  DB1G
           NUMBER OF RECORDS PROCESSED WITHOUT A CORRELATION HEADER WAS 9

DGOC4005I  DSNDB0G           DSNDB0G   DB2G  DB2G
           NUMBER OF RECORDS PROCESSED WITHOUT A CPU HEADER WAS 30

DGOC4010I  DSNDB0G           DSNDB0G   DB2G  DB2G
           NUMBER OF RECORDS PROCESSED WITHOUT A CORRELATION HEADER WAS 9
```

Figure 47. Job Summary Log

The job summary log can be useful, for example, when you want to see what traces were started and stopped during the reporting period.

Chapter 8. Exception Reporting

Exception reporting is the most effective way of using DB2 PM to identify 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.

Exception reporting identifies DB2 threads and statistics intervals with fields containing values outside the limits you have specified. This helps you manage your performance objectives by highlighting problems in the DB2 subsystem:

- Applications that are experiencing exceptional conditions in the DB2 subsystem
- Exceptional DB2 subsystem conditions that are causing thread performance problems.

The limits, called 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 containing a value outside the limits you have specified are reported.

Output

The following types of exception processing output are available in accounting and statistics:

- Reports
- Traces
- File data sets
- Logs
- Log file data sets.

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.

Exception logs combine both accounting and statistics fields that are in exception status into one report and show the information in timestamp order.

The exception file data set and the exception log file data set are sequential data sets suitable for use by the DB2 load utility. The file data set contains either accounting or statistics *records* that have at least one field in exception status and the log file data set contains both accounting and statistics *fields* that are in exception status. In other words, the file data set corresponds to an exception trace, whereas the log file data set corresponds to an exception log.

Exception reports, traces, and files are obtained using the EXCEPTION option in the accounting and statistics subcommands.

The exception log is generated for a DB2 PM execution when the EXTRCDD1 DD statement is defined in your JCL and the exception log file data set is generated when the EXFILDD1 DD statement is defined in your JCL.

You can also generate an exception log file data set using the Online Monitor. Refer to the *DB2 PM Online Monitor User's Guide* for more information.

Specifying Exceptions

Exception threshold values are defined in the exception threshold data set. You can specify these thresholds using the Exception Threshold Field Details panel, or calculate the thresholds using exception profiling. When exception processing is active, the DB2 instrumentation data is checked against values in the exception threshold data set. Fields containing values outside limits that you have specified are reported.

To get you started, the sample threshold member DGOETV41 has been provided in SDGODATA.

This member is a sample only and can be modified as required by your installation.

Note that some of the entries in this member contain asterisks (*) in their exception threshold fields. Values are substituted for these asterisks when exception profiling is performed. If you use this sample data set before performing exception profiling, the entries containing asterisks generate warning messages and are not processed, however, all other entries are processed in the normal manner.

If you want to use a new exception threshold data set, it should be allocated using the following attributes:

RECFM:	VB
LRECL:	255
BLKSIZE:	5 104

Note: As the sample threshold member uses a record length that differs from the record length of the exception threshold data set, you are alerted that records are truncated when you copy DGOETV41 to your newly allocated data set. In this case you can ignore this warning.

The Exception Threshold Field Details panel is accessed indirectly via the Data Set Maintenance Menu (DGOPMENU). To display this menu (Figure 48), select option 4 *Maintain parameter data sets* from the DB2 PM 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
'DGO.V6R1M0.THRESH'_______________________

DPMPARMS data set
'DGO.V6R1M0.DPMPARMS'_____________________






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

Figure 48. Selecting Maintain Exception Threshold Data Set

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

Press **Enter**. The Exception Threshold Category Selection panel is displayed, as shown in Figure 49.

```
DGOPXDS1           Exception Threshold Category Selection
Command ===> ___________________________________________________________

Select one or more categories, then press Enter.  Overtype with space to
deselect any category.  Request EXIT when complete.

    Category
/   Elapsed, CPU and Waiting Times per Plan Execution
_   Elapsed, CPU and Waiting Times per Program Execution
_   CPU Times per Address Space
_   SQL Statements per Plan Execution
_   SQL Statements per Program Execution
_   SQL Statements per System
_   Subsystem Events per Plan Execution
_   Subsystem Events per System
_   Locking Activity per Plan Execution
_   Locking Activity per System
_   RID List Processing per Plan Execution
_   RID List Processing per System
_   Query Parallelism per Plan Execution
_   Query Parallelism per System
_   Buffer Pools Activity per Plan Execution
_   Buffer Pools Activity per System
_   Distributed Activity per Location per Plan Execution
_   Distributed Activity per System
_   Distributed Activity per Location per System
_   IFI and Data Capture Activity per Plan Execution
_   IFI Activity per System
_   EDM Pool Activity per System
_   Open/Close Activity per System
_   Plan/Package Processing per System
 F1=Help    F2=Split   F3=Exit    F7=Up      F8=Down    F9=Swap   F12=Cancel
```

Figure 49. Exception Threshold Category Selection Panel

Exception Reporting

This panel displays the categories of exception threshold fields you can choose from. The name of the category indicates the area where the exception applies: *per plan* or *per program* exceptions are reported in accounting and *per system* or *address space* exceptions are reported in statistics.

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

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

In Figure 49 on page 97, the *Elapsed, CPU and Waiting Times per Plan Execution* field is selected. When you press **Enter**, the Exception Threshold Field Selection panel is displayed showing all fields associated with this category, as shown in Figure 50.

```
DGOPXDS2           Exception Threshold Field Selection
Command ===> ____________________________________________________________

Select one or more fields, then press Enter.  Overtype with space to
deselect any field.  Request EXIT when complete.

Field category  . . : Elapsed, CPU and Waiting Times per Plan Execution

   Field     Description
>  ADCPUT    CPU time in application (Class 1)
/  ADRECETT  Elapsed time in application (Class 1)
_  ADTWTAP   Total wait time in application (Class 1)
_  ADDB2ETT  Elapsed time in DB2 (Class 2)
_  ADDBBCPUT CPU time in DB2 (Class 2)
_  ADTWTDB   Total wait time in DB2 (Class 2)
_  ADTSUST   Total Class 3 suspensions time
_  ADTSUSC   Total Class 3 suspensions
_  QWACAWTL  Lock/latch suspensions time (Class 3)
_  ADLLSUSC  Lock/latch suspensions (Class 3)
_  QWACAWTI  Synchronous I/O susp. time (Class 3)
_  ADIOSUSC  Synchronous I/O suspensions (Class 3)
_  QWACAWTR  Other read I/O susp. time (Class 3)
_  ADARSUSC  Other read I/O suspensions (Class 3)
_  QWACAWTW  Other write I/O susp. time (Class 3)
_  ADAWSUSC  Other write I/O suspensions (Class 3)
_  QWACAWTE  Serv.task switch susp. time (Class 3)
_  ADSTSUSC  Serv.task swtch suspensions (Class 3)
-- End of Items --

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

Figure 50. Exception Threshold Field Selection Panel

This panel displays all fields in this category. Fields that have been selected previously are marked with a greater-than symbol (>) and sorted to the top.

Select a field to specify the exception thresholds for it. Type a slash ('/') or 'S' in the selection field and press **Enter**. The Exception Threshold Field Details panel is displayed (Figure 51).

```
DGOPXDSN           Exception Threshold Field Details
Command ===> ______________________________________________________________

                                                              ENTRY 1 OF 1

Category  . . . . . : Elapsed, CPU and Waiting Times per Plan Execution
Field ID  . . . . . : ADRECETT
Description . . . . : Elapsed time in application (Class 1)



Active  . . . . . . . . . . . . . 1  1=Yes   2=No

By  . . . . . . . . . . . . . . . 1  1=Total   2=Minute   3=Second
                                     4=Commit  5=Thread

Compare operator  . . . . . . . . <  <=Less than >=Greater than
Warning threshold . . . . . . . . 4
Problem threshold . . . . . . . . ____________

Local location  . . . . . . . . . *
Group name  . . . . . . . . . . . *
Subsystem ID  . . . . . . . . . . *
Member name . . . . . . . . . . . *
Requester location  . . . . . . . *
Connect . . . . . . . . . . . . . *
Planname  . . . . . . . . . . . . *
Corrname  . . . . . . . . . . . . *
Corrnmbr  . . . . . . . . . . . . *
Primauth  . . . . . . . . . . . . *
 F1=Help      F2=Split     F3=Exit      F5=Add       F6=Delete    F7=Up
 F8=Down      F9=Swap     F10=Previous F11=Next     F12=Cancel
```

Figure 51. Exception Threshold Field Details

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 using different criteria.

The first three fields on 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. Specify 2 (No), if you do not want to use the exception field specification when you next generate exception reports, but you do not want to delete the entry.
- 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 comparison *operator*. It can be greater than or less than the threshold.
- The thresholds. You can specify two kinds of thresholds for a field: *warning* and *problem* thresholds. 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 when 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 on this panel, press **F5**, and specify a different connection name on 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 makes sense 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, the threshold is checked for each thread if you specify *Total*. If you specify *By Thread*, then 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*, then DB2 PM 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*, then 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*, then 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.

Example of Producing an Accounting Exception Report

The following 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 like the example shown in Figure 52 and that the report would normally be fairly long, over ten pages.

```
   LOCATION: DSNDB0G                          DB2 PERFORMANCE MONITOR (V6)                          PAGE: 1-1
      GROUP: DSNDB0G                            ACCOUNTING REPORT - SHORT                 REQUESTED FROM: NOT SPECIFIED
     MEMBER: DB1G                                                                                    TO: NOT SPECIFIED
  SUBSYSTEM: DB1G                                ORDER: PRIMAUTH-PLANNAME                  INTERVAL FROM: 03/19/99 22:35:07.26
DB2 VERSION: V6                                      SCOPE: MEMBER                                   TO: 03/19/99 22:56:45.89

PRIMAUTH                    #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
 PLANNAME                   #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
--------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

SYSADM                           20       0    0.00    0.00    0.00    0.00       0.253473       0.253207      N/P      N/P     0.00
 'BLANK'                          0      20    0.00    0.00    0.00    0.00       0.009828       0.009540      N/P      N/P        0

USRT002                           7       0    0.00    0.00    1.00    0.00    1:32.664960    1:32.601455 32711.57     8.57    18.14
 DSNTEP61                         0       7    1.14    0.57    0.57    1.57      46.990679      46.971450 38572.71  1018.14        0
⋮

⋮
```

Figure 52. Accounting Report—without Exception

If you wanted to use this report to find out, for example, 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 yourself 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. We assume that the connection ID for the CICS address space is CICSA.

Access DGOETV41 in SDGODATA through the IRF and specify the thresholds in the Exception Threshold Field Details panel.

```
DGOPXDSN           Exception Threshold Field Details
Command ===> ______________________________________________________________

                                                              ENTRY 1 OF 1

Category  . . . . . : Elapsed, CPU and Waiting Times per Plan Execution
Field ID  . . . . . : ADRECETT
Description . . . . : Elapsed time in application (Class 1)



Active  . . . . . . . . . . . . . 1  1=Yes   2=No

By  . . . . . . . . . . . . . . . 5  1=Total   2=Minute   3=Second
                                     4=Commit  5=Thread

Compare operator  . . . . . . . . >  <=Less than >=Greater than
Warning threshold . . . . . . . . 3
Problem threshold . . . . . . . . 6

Local location  . . . . . . . . . *
Group name  . . . . . . . . . . . *
Subsystem ID  . . . . . . . . . . *
Member name . . . . . . . . . . . *
Requester location  . . . . . . . *
Connect . . . . . . . . . . . . . CICSA
Planname  . . . . . . . . . . . . *
Corrname  . . . . . . . . . . . . *
Corrnmbr  . . . . . . . . . . . . *
Primauth  . . . . . . . . . . . . *
 F1=Help      F2=Split     F3=Exit      F5=Add       F6=Delete    F7=Up
 F8=Down      F9=Swap     F10=Previous F11=Next     F12=Cancel
```

Figure 53. Specifying Exceptions

In the above example we have specified that:

- Exceptions will be checked for the class 1 elapsed time field ADRECETT.
- The elapsed time thresholds are checked for the average value per thread, that is, the value in this field is divided by the number of threads.
- The comparison is *greater than*.
- The class 1 elapsed time value will be flagged as a warning exception if it exceeds 3 seconds per thread and as a problem exception if it exceeds 6 seconds per thread.
- 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.

Exception Reporting

Now you can run the accounting exception report. Remember to specify the name of your exception threshold data set for the EXCPTDD ddname in your JCL. Use the following command to produce the report:

```
⋮

ACCOUNTING
      REPORT
         EXCEPTION
⋮
```

This is what the accounting exception report looks like:

```
   LOCATION: DSNDB0G                            DB2 PERFORMANCE MONITOR (V6)                             PAGE: 1-1
      GROUP: DSNDB0G                              ACCOUNTING REPORT - SHORT                    REQUESTED FROM: NOT SPECIFIED
     MEMBER: DB1G                                         EXCEPTION                                       TO: NOT SPECIFIED
  SUBSYSTEM: DB1G                                 ORDER: PRIMAUTH-PLANNAME                      INTERVAL FROM: 03/19/99 22:35:07.26
DB2 VERSION: V6                                       SCOPE: MEMBER                                       TO: 03/19/99 22:56:45.89

PRIMAUTH                   #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
 PLANNAME                  #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
-------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

SYSADM                          20       0    0.00    0.00    0.00    0.00       0.253473       0.253207      N/P      N/P     0.00
 'BLANK'                         0      20    0.00    0.00    0.00    0.00       0.009828       0.009540      N/P      N/P        0

  *********************************************************************************************************************************
  *  TYPE          FIELD ID   FIELD DESCRIPTION                                    BY                     VALUE   THRESHOLD     *
  *                           FIELD QUALIFIER                                                                                   *
  *  WARNING       ADRECETT   ELAPSED TIME IN APPLICATION (CLASS 1)                TOTAL               5.069463 > 0             *
  *                                                                                                                             *
  *********************************************************************************************************************************
```

Figure 54. Accounting Exception Report (Part 1 of 2)

```
   LOCATION: DSNDB0G                            DB2 PERFORMANCE MONITOR (V6)                             PAGE: 2-1
      GROUP: DSNDB0G                              ACCOUNTING REPORT - SHORT                    REQUESTED FROM: NOT SPECIFIED
     MEMBER: DB2G                                         EXCEPTION                                       TO: NOT SPECIFIED
  SUBSYSTEM: DB2G                                 ORDER: PRIMAUTH-PLANNAME                      INTERVAL FROM: 03/19/99 22:47:05.09
DB2 VERSION: V6                                       SCOPE: MEMBER                                       TO: 03/19/99 22:47:05.09

PRIMAUTH                   #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
 PLANNAME                  #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
-------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

USRT002                          1       0    0.00    0.00    1.00    0.00    8:59.989315    8:59.871297 45769.00    43.00    62.00
 DSNTEP61                        0       1    0.00    0.00    0.00    1.00    2:47.027856    2:47.010331   270.0K  1429.00        0

  *********************************************************************************************************************************
  *  TYPE          FIELD ID   FIELD DESCRIPTION                                    BY                     VALUE   THRESHOLD     *
  *                           FIELD QUALIFIER                                                                                   *
  *  WARNING       ADTDDL     TOTAL SQL DDL STATEMENTS                             TOTAL                      0 < 1             *
  *                                                                                                                             *
  *********************************************************************************************************************************
⋮

⋮
```

Figure 54. Accounting Exception Report (Part 2 of 2)

The exception report is much shorter than the original accounting report. It only lists transactions that have a class 1 elapsed time that has exceeded the exception threshold and flags them as warnings or problems.

If you had specified more than one exception threshold, all records that contained any fields reaching an exception threshold value would be listed.

Which Exception Fields to Choose

DB2 PM provides a comprehensive set of exception fields and it can be hard to decide which ones to choose for your site. As a rule, most sites would only need to define a limited number of thresholds.

In general, long response times are a good indicator of a performance problem and therefore you should always start by defining exception thresholds for time fields.

To make it easier for you to decide which exception fields to choose, exception threshold data set DGOETV41 is provided in SDGODATA. This member is only an example, and you can modify it to suit your installation.

To use exception processing efficiently, you must also consider what are the most important applications or transactions in your system. You should always define exception thresholds for critical business applications, whereas you do not need to monitor less important applications as closely. Frequently executed applications are good candidates for exception thresholds, seldom run applications less so.

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 should consider carefully the fields for which to specify exception thresholds. The more fields you specify, the greater the impact on processing.

What Values to Specify

The exception threshold member supplied in SDGODATA has predefined warning and problem values specified for some fields, but not for all of them. This is because the values for some of the fields, such as greater than zero for the number of deadlocks, apply to most sites, whereas others, such as elapsed times, vary according to site and application. These fields are marked with a threshold value of asterisk (*) in the sample data sets.

If you know what values to specify for these site-specific thresholds, fill them in. The performance objectives stated in your service level agreement are a good starting point. Accounting TOP lists are also helpful in determining the threshold values. You can also modify the predefined threshold values and specify additional exception fields.

If you are not sure what values to specify, use the exception profiling function to fill in these values for you. Mark all the thresholds you want exception profiling to determine with an asterisk. For more information, refer to “Tailoring Exception Thresholds” on page 113.

Chapter 9. Streamlining DB2 PM Processing

The best way to streamline DB2 PM processing is to ask only for the information you really need.

DB2 PM is a comprehensive reporting tool, but for the daily monitoring of DB2 you need a very limited amount of information. If you request a detailed report using all your input data gathered over a long period of time, DB2 PM processing takes up a lot of system resources and you end up with pages of information you are probably not interested in.

So, to avoid unnecessary performance overhead and to save time:
- Filter the input data, preferably using GLOBAL
- Disable DB2 PM internal sort if appropriate
- Consider carefully how detailed a report you need
- Define groups for identifiers 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 you are interested in
- Specify DPMOUT or keep a save data set only if you are sure you want to report the data again
- Limit the number of SQL statements you want explained.

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 whenever you can, because only the data that passes the GLOBAL filters is processed further. The less data DB2 PM needs to process, the better the performance.

You should, however, make sure that you do not exclude records needed in DB2 PM processing. For more information, refer to “Database and Page Set Names Not Translated” on page 173 and the description of GLOBAL in the *DB2 PM Report Reference*.

FROM and TO

The simplest filter is the start and end time of the data to be reported. Specify the start and end times using the FROM and TO options.

Assuming that 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, then you could specify the following:

```
⋮

GLOBAL
     FROM (03/25/98,08:30)
     TO   (03/25/98,17:00)
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.

Presuming that your input data set contains data for more than one day, for example a week, you can generate reports that show the performance of your system during peak hours for the whole week by specifying:

```
⋮

GLOBAL
     FROM (,08:30)
     TO   (,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 every day of the week.

If you wanted to generate a lockout report for a day in the same job step, you could enter the following:

```
⋮

GLOBAL
     FROM    (,08:30)
     TO      (,17:00)
ACCOUNTING
     REPORT
STATISTICS
     TRACE
LOCKING
     REPORT
        FROM (03/27/98)
        TO   (03/27/98)
⋮
```

Note that 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 DB2 PM identifier values, for example, user IDs or plans. You can do this using the INCLUDE and EXCLUDE options.

DB2 PM Identifiers

The identifiers describe the objects DB2 PM reports on. The most commonly used DB2 PM identifiers describe:

- **The location**

  The *LOCATION* identifier is the name of the DB2 subsystem. If your input data contains data from several locations, you can include data only for those locations you are interested in.
- **The user**

  DB2 PM uses two identifiers for the user ID. The first one is the value of the authorization ID at the time of connection to DB2 (*ORIGAUTH*) and the second one is the authorization ID set at signon or identify (*PRIMAUTH*).

  For more information, refer to “Appendix C. Comparing Original Authorization IDs with Primary Authorization IDs” on page 275.

  For SQL requests from a client, the user ID of the end user at the workstation is a possible identifier as well.
- **The plan**

  Use the *PLANNAME* identifier to select specific plans. Some examples 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, and MAINPACK to define plans that contain a specific package. For more information, refer to “Defining the MAINPACK Identifier” on page 133.
- **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, 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, refer to “Correlation ID Translation” on page 132.
- **Distributed activity**

  When you are reporting data for distributed processing you can select the locations requesting 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).

Examples

For example, if your input data contains data from more than one location, but you only want to inspect the performance of location USIBMSNEWY11, specify the following:

```
⋮
GLOBAL
     INCLUDE (LOCATION(USIBMSNEWY11))
⋮
```

Only data for location USIBMSNEWY11 is processed.

Perhaps you are not interested in the CICS activity for that location. Therefore you would specify:

```
⋮
GLOBAL
     INCLUDE (LOCATION(USIBMSNEWY11))
     EXCLUDE (CONNTYPE(CICS))
⋮
```

The only location reported is USIBMSNEWY11 and all other connections are reported except CICS.

Or assume that you suspect that authorization identifier USERID01 using plan NEWACC is causing a performance problem. Therefore you want to report only data belonging 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 DB2 PM Internal Sort

When requesting accounting functions only, it is often possible to avoid the DB2 PM internal sort of the input data. This reduces the size of the sort work files which 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 when the input data set has been sorted, for example when reprocessing data from the DPMOUT data set.

ACCEPT Disable the sort and accept out-of-sequence records. Using this option it is possible to create complete and accurate accounting reports (including records from multiple locations) from data sets which 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. Take care to avoid the inclusion of duplicate records through the concatenation of two data sets which contain the same trace records, for example if trace data has been collected on two data sets during the same period.
- Only one location is reported per trace, to report more, specify multiple TRACE subcommands with INCLUDE on location.
- Trace entries might not be printed in time sequence. If a trace contains entries which are out of sequence, a message appears at the end showing the number.
- When INTERVAL(0) is in effect (the default), the interval times appearing 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 Amount of Detail

Before producing a report or a trace, consider how much detail you need.

Use the LAYOUT option of the accounting and statistics reports and traces to control the amount of data to be produced. Do not use the most comprehensive layouts showing all the possible fields unless you specifically need to. In most situations the default layouts, which are short versions of reports and traces, provide enough detail for monitoring and problem determination.

Another way of reducing the amount of data to be reported is to tailor your own report layouts according to your needs (see “Tailoring Report Layouts” on page 115). In this way you can avoid processing data you are not interested in, focus your attention on meaningful information, and make the most efficient use of DB2 PM.

Group Data

Use the GROUP command to define a set of DB2 PM identifier values that can be used when requesting certain reports. The information for the set is reported as a single entry on the reports.

In GROUP processing the data for all members of the set is consolidated into one record. This improves DB2 PM performance because fewer records need to be processed.

Streamlining Processing

Sets are also useful for reporting purposes, for example, when you want to report data for an entire department instead of every individual member in it.

This is how sets can be specified:

Let's say that your sales department consists of three users, USER001, USER002, and USER003, and that you want to produce an accounting report showing 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 a DB2 PM 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 DB2 PM performance, but it can make it easier to specify and read your command stream.

You might want a breakdown on how some plans used by the sales department affect performance. To do that, you want to produce an accounting report and an SQL activity trace. If you specify a list for all the plan name values as in the example below, you can use the list name (SALES) in all the 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 the individual plans specified in the list.

You can use LIST with all the report sets. Lists can be specified for any DB2 PM identifier values allowed in that report set.

Be Careful with INTERVAL

If you want to use DB2 PM 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 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.

You should, however, bear in mind that interval processing affects the performance of the DB2 PM job. So, if you do not intend to produce reports showing the performance of your system at time intervals, use 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 find out whether there are problems in your system performance. You can set exception thresholds for virtually all the accounting and statistics fields. You should, however, carefully consider the fields for which to specify exception thresholds. The more fields you specify, the greater the impact on processing.

So, specify exception thresholds only for those fields that you believe will signal poor performance in your environment. For information on how to generate exception reports and for a list of recommended threshold fields, refer to “Chapter 8. Exception Reporting” on page 95.

Do You Need a DPMOUT Data Set?

You can keep the preprocessed input data in the DB2 PM output data set DPMOUT if you wish to do so. The DPMOUT data set can be used as input to DB2 PM.

If you do not specify DPMOUTDD, only the records required for the current job step are processed, which improves DB2 PM performance.

Do You Want to Save Reduced Data?

If you need to keep historical accounting and statistics data about DB2 performance, you might 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 data set depends on the reduction interval you have 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.

The save data sets are also needed as input to accounting and statistics graphs. For more information, refer to “Chapter 13. Producing Graphs” on page 157.

See also “Appendix B. Using GROUP to Improve Save Data Set Performance” on page 273.

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 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 from.

Chapter 10. Customizing DB2 PM Functions

There are some DB2 PM features that you might want to customize to meet your particular needs. The things you can tailor are:

- *Exception Thresholds*

  Exception reporting identifies DB2 threads and statistics intervals with fields containing values outside limits, exception thresholds, you have specified. You can either specify the thresholds yourself (for more information refer to "Chapter 8. Exception Reporting" on page 95) or you can let DB2 PM fill these values.
- *Report layouts*

  You can tailor accounting and statistics report layouts, if none of the supplied model reports provide the information in the format you want.
- *Time zone specifications*

  You might want to 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*

  Even though DB2 PM provides a default translation for most environments, you might want to change the way DB2 PM translates the DB2 correlation identifier into correlation name and number.
- *MAINPACK identifier*

  MAINPACK is a DB2 PM 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 descriptions*

  You can modify exception field descriptions using the ISPF editor.

Any changes you make to the above features are recorded in the DPMPARMS data set. The data set needs to be allocated; refer to the *DB2 PM Report Reference* for information about the attributes. If you decide to use the model layouts and default specifications, you do not need to specify DPMPARMS. Note that there is no limit to how many DPMPARMS data sets you create and that you can concatenate several DPMPARMS data sets.

Tailoring Exception Thresholds

If you are not sure what values to specify for exception thresholds, use the DB2 PM exception profiling function to fill in these values for you. Mark all the thresholds you want exception profiling to determine with an asterisk.

Exception profiling is a DB2 PM batch job that sets thresholds in an exception threshold data set.

To use this function, access the IRF and select option 6, *Exception Profiling* from the DB2 PM main menu. The Exception Profiling panel is displayed.


```
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 55. Exception Profiling Panel

On this panel:

- Specify the percentage of input data you want to be flagged as warnings and the percentage of input data you want flagged as problems.
- Choose whether you want to produce a profile report. The profile report documents the results of exception profiling showing the expected number of exceptions for various thresholds.
- Specify the name of the input data set containing data from your DB2 subsystem. It can be a GTF, SMF, or DPMOUT data set.

  If you want to use multiple input data sets, you can concatenate them by editing the generated job stream using option 2, *Edit the generated job stream*, on the Job Processing Selections panel (see Figure 56).

  The records in this data set should be representative of the type of data 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.
- Specify the name of the input exception threshold data set that contains entries for the fields you want checked. It can be the exception threshold data set provided in SDGODATA or your own data set. Mark the fields for which you want the program to provide threshold values with an asterisk.
- Specify the name of the output exception threshold data set that will contain the threshold values.
- Specify the data set information for the profiling report.

After you have completed the specifications, press **Enter** to generate the exception profiling job stream. The Job Processing Selections panel (DGOOJOBM) is displayed.

```
DGOOJOBM                    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:
//USERP01P 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 56. 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. It contains a copy of the input threshold data set together with the threshold values DB2 PM has determined using the input data and the profiling criteria.

Check the profiling report to make sure that the exception thresholds and the number of exceptions are satisfactory. If you need to, you can modify the thresholds using the data set editor.

Now you can generate exception reports using the threshold data set created by the exception profiling function.

Tailoring Report Layouts

Read this section only if you want to modify accounting and statistics model layouts.

DB2 PM supplies a number of accounting and statistics model reports and traces, but if none of them suit the requirements at your site, you can tailor your own layouts using the user-tailored reporting feature (UTR). You can create as many layouts as you want.

Some examples of model layouts are shown in “Chapter 18. Monitoring the Subsystem—Statistics” on page 227 and “Chapter 16. Monitoring Applications—Accounting” on page 179. For a description of all layouts, refer to the appropriate chapters in the *DB2 PM Report Reference*.

Customizing Functions

User-tailored reporting gives you full control over the volume, contents, and layout of accounting and statistics traces and reports. You can:

- Add entire blocks and individual fields to an existing layout.

  You might want to include some additional fields, or entire blocks of related fields, in a trace or report.
- Remove entire blocks and individual fields from an existing layout.

  You might want to produce more compact traces or reports by excluding some fields or entire blocks of related fields which are not of interest to you.
- Change the relative positions of blocks and fields in an existing layout.

  By rearranging blocks and fields, you can produce more compact traces or reports or make fields that are of particular interest to you more prominent.
- Change block and field labels.

  You can define the block and field labels and use abbreviations you find meaningful. You can choose to either spell out the block or field names or use acronyms and abbreviations to keep reports compact.

Tailoring an Accounting Report

The following example shows how to add a block of fields to an existing report layout and replace some of the existing fields with new ones.

Suppose that the layout of the short accounting report (supplied as accounting report layout SHORT) does not report all the information you need in order to monitor your DB2 installation and its specific workload. The standard layout, generated with the default command options, looks like this:

```
   LOCATION: STLEC1                                   DB2 PERFORMANCE MONITOR (V6)                              PAGE: 1-1
      GROUP: DSNCAT                                      ACCOUNTING REPORT - SHORT                     REQUESTED FROM: NOT SPECIFIED
     MEMBER: SSDQ                                                                                                 TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                         ORDER: PRIMAUTH-PLANNAME                      INTERVAL FROM: 04/06/98 20:18:00.23
DB2 VERSION: V6                                               SCOPE: MEMBER                                       TO: 04/06/98 20:48:38.68

PRIMAUTH                    #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
 PLANNAME                   #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
--------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

ADMF001                          13       0    0.00    0.00    0.00    0.00       0.662349       0.662264      N/P      N/P     0.00
 'BLANK'                          0      13    0.00    0.00    0.00    0.00       0.004677       0.004589      N/P      N/P        0

ADMF001                           3       2    0.00    0.00    0.00    0.00       5.161409       5.161304   220.67     9.00     0.00
 DSNBIND                          0       3    0.00    0.00    0.00    0.00       0.139584       0.139478   157.00     0.67        0
⋮

⋮
```

Figure 57. Accounting Report—Sample Layout

Assume that buffer pool utilization is the most critical part of the system activity at your site and therefore you want to include more buffer pool information in your layout. The report you want to generate looks like this:

```
   LOCATION: STLEC1                           DB2 PERFORMANCE MONITOR (V6)                              PAGE: 1-1
      GROUP: DSNCAT                            ACCOUNTING REPORT - BUFFER                   REQUESTED FROM: NOT SPECIFIED
     MEMBER: SSDQ                                                                                      TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                               ORDER: PRIMAUTH-PLANNAME                     INTERVAL FROM: 04/06/98 20:18:00.23
DB2 VERSION: V6                                     SCOPE: MEMBER                                        TO: 04/06/98 20:48:38.68

PRIMAUTH                    #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME      CL3 SUSP LOCK SUS
 PLANNAME                   #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME CLS3 SUSP.TIME #LOCKOUT
--------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------------- --------

ADMF001                          13       0    0.00    0.00    0.00    0.00       0.662349       0.662264          7.77     0.00
 'BLANK'                          0      13    0.00    0.00    0.00    0.00       0.004677       0.004589      0.537943        0

ADMF001                           3       2    0.00    0.00    0.00    0.00       5.161409       5.161304         39.00     0.00
 DSNBIND                          0       3    0.00    0.00    0.00    0.00       0.139584       0.139478      2.472389        0

    ------------------------------------------------------------------------------------------
    |BP ID     GETPAGES  SYN.READ  SEQ.PREF  LISTPREF  DYN.PREF  ASY.READ  BUF.UPDT  #SYNWRTE|
    |BP0         220.67      9.00      0.00      0.00      0.67      0.67    157.00         0|
    ------------------------------------------------------------------------------------------

ADMF001                           9       0    0.00    4.22    1.00    0.00      30.714273      30.067720        141.13     0.00
 DSNTEP61                         0      15    3.67    0.56    0.56   10.22       0.346763       0.305899     23.810092        0

    ------------------------------------------------------------------------------------------
    |BP ID     GETPAGES  SYN.READ  SEQ.PREF  LISTPREF  DYN.PREF  ASY.READ  BUF.UPDT  #SYNWRTE|
    |BP0         360.00     99.56      0.89      0.00      0.00      6.44     99.56         6|
    |BP1           2.00      0.00      0.00      0.00      0.00      0.00      0.00         0|
    |BP2           6.22      0.33      0.22      0.00      0.00      0.89      3.56         2|
    |TOT4K       368.22     99.89      1.11      0.00      0.00      7.33    103.11         8|
    ------------------------------------------------------------------------------------------
⋮
⋮
```

Figure 58. Accounting Report—Tailored Layout

The tailored layout has been named BUFFER and contains buffer pool information in addition to the general accounting data. Note that the buffer pool fields in the general accounting data block have been replaced with other fields in order not to duplicate information. In this example we have chosen to show class 3 suspension times and events.

The following steps show you how to produce the tailored layout.

Selecting a Layout

Access the IRF. The DB2 PM main menu is displayed.

```
DGOFMENU           IBM Database 2 Performance Monitor

Select one of the following.

5_ 1. Create and execute DB2 PM commands
   2. Display and print graphs
   3. View online DB2 activity
   4. Maintain parameter data sets
   5. Customize DB2 PM report and trace layouts
   6. Exception profiling



 IBM DB2 UDB Performance Monitor for OS/390 V6
 Licensed Materials - Property of IBM
 5645-DB2 (C) Copyright IBM Corp. 1985, 1998.
 All rights reserved.
 US Government Users Restricted Rights -
 Use, duplication or disclosure restricted
 by GSA ADP Schedule Contract with IBM Corp.

Command ===> ___________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F6=History  F9=Swap      F12=Cancel
F17=Collect
```

Figure 59. DB2 PM Main Menu—Select Tailor Layouts

Select *Customize DB2 PM report and trace layouts* from the DB2 PM main menu by entering 5 in the input field.

The User-Tailored Reporting Layout Generation panel is displayed.

```
DGOBMENU         User-Tailored Reporting Layout Generation


Select one of the following report set functions and then enter the
DPMPARMS data set to be used.

1_ 1. Accounting report
   2. Statistics report
   3. Accounting trace
   4. Statistics trace




DPMPARMS data set
'DGO.V6R1M0.MYMPARMS'____________________________






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

Figure 60. Selecting Accounting Report

To change the accounting report layouts, select option 1 from the menu, enter the name of the DPMPARMS data set where the customized layout is to be stored, and press **Enter**.

The UTR Layout Selection panel is displayed.

```
DGOBPLAY                  UTR Layout Selection               ROW 1 TO 2 OF 2

Report set  . . . . . : Accounting Report

Select a Layout then press Enter.

    Layout  Description
_   LONG    Accounting Report - Long
/   SHORT   Accounting Report - Short
******************************* BOTTOM OF DATA ********************************











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

Figure 61. Selecting a Model Layout

Select layout SHORT to be used 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 closest to the layout you want.

The UTR Block Selection panel for the Accounting Report SHORT layout is displayed.

Adding a Block

```
DGOBPBLK                  UTR Block Selection              Row 1 to 18 of 27

Report set  . . . . . : Accounting Report
Layout  . . . . . . . : SHORT

1.  To change the contents of a block, select the Modify column
2.  To include and order, modify/add a number in the Order column
3.  To exclude a block, blank out the Order column

Modify  Order  Label
_       10     GENERAL
_       20     PACKAGE GENERAL
_       30     DISTRIBUTED ACTIVITY
_       99999  ORDER IDS
_       99999  AVERAGE
_       99999  CLASS 3 SUSP.
_       99999  HIGHLIGHTS
_       99999  SQL DML
_       99999  SQL DCL
_       99999  SQL DDL
_       99999  LOCKING
_       99999  NORMAL TERM.
_       99999  ABNORMAL TERM.
_       99999  IN DOUBT
_       99999  DRAIN/CLAIM
_       99999  DATA CAPTURE
_       99999  DATA SHARING
_       99999  QUERY PARALLELISM

Command ===> ______________________________________________________________
 F1=Help    F2=Split   F3=Exit    F5=Extend  F6=Browse  F7=Up      F8=Down
 F9=Swap   F12=Cancel
```

Figure 62. Block Selection Panel

The *Order* column shows that the blocks included in the current layout are the blocks with labels GENERAL, PACKAGE GENERAL, and DISTRIBUTED ACTIVITY (the last two blocks are printed only if the information is present in the input data). The numbers in the *Order* column indicate the sequence in which the blocks are printed. Note that fields that have not been selected are numbered 99999.

To add the buffer pool block to the layout, 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 Figure 63. In this example, the column form was chosen.

```
DGOBEBLK                    UTR Block Selection           Row 20 to 27 of 27

Report set  . . . . . : Accounting Report
Layout  . . . . . . . : SHORT

1.  To change the contents of a block, select the Modify column
2.  To include and order, modify/add a number in the Order column
3.  To exclude a block, blank out the Order column

Modify  Order  Label
               Description
_       99999  RID LIST
               RID List Activity Data
_       99999  BUFFER POOL ACTIVITY
               Buffer Pool Activity Data (Table Form)
_       99999  GROUP BUFFER POOL
               Group Buffer Pool Activity Data (Table Form)
_       99999  DISTRIBUTED ACTIVITY
               Distributed Data Facility (List Form)
_       99999  RESOURCE LIMIT FACILITY
               Resource Limit Facility Data (Column Form)
_       99999  PACKAGE ACTIVITY
               Package Activity Data
_       99999  IFI CLASS 5
               IFI (Class 5) times
/          15  BUFFER POOL ACTIVITY
               Buffer Pools Activity Data (Column Form)
****************************** Bottom of data ********************************

Command ===> __________________________________________________________
 F1=Help    F2=Split   F3=Exit    F5=Extend  F6=Browse  F7=Up      F8=Down
 F9=Swap   F12=Cancel
```

Figure 63. Adding a Block

Enter any number between 10 and 20 in the *Order* column next to the block label.

The buffer pool block will now be printed after the general accounting data block and before the general package data (if any) block. If you wanted the block to be printed before the general accounting block, you would specify a number less than 10.

To see which fields can be included in the block, place a selection character (/) in the *Modify* column next to the block name.

The UTR Field Selection panel is displayed.

```
DGOBFLD4                    UTR Field Selection            ROW 1 TO 9 OF 17

Report set  . . . . . : Accounting Report
Layout  . . . . . . . : SHORT
Block label . . . . . : BUFFER POOL ACTIVITY

1.  To include and order, modify/add a number in the Order column
2.  To exclude a field, blank out the Order column

Order  Label
       Description                                                    Length
10     BP ID
       Buffer Pool ID (QBACPID)                                            8
20     GETPAGES
       Average number of getpage requests (QBACGET)                        8
30     SYN.READ
       Average number of synchronous reads (QBACRIO)                       8
40     SEQ.PREF
       Average number of sequential prefetches (QBACSEQ)                   8
50     LISTPREF
       Average number of list prefetches (QBACLPF)                         8
60     DYN.PREF
       Average number of dynamic prefetches (QBACDPF)                      8
70     ASY.READ
       Average of pages read asynchronously (QBACSIO)                      8
80     BUF.UPDT
       Average number of buffer updates (QBACSWS)                          8
90     #SYNWRTE

Command ===> __________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F5=Extend    F6=Qualify   F7=Up
 F8=Down      F9=Swap     F10=Config   F11=Filler   F12=Cancel
```

Figure 64. Accounting Report—Buffer Pools Activity

You can press **F5** to hide the label descriptions and field lengths. To see all the available fields, scroll forward pressing **F8**. As in the Block Selection panel, selected fields have an order number in front of them and fields that have not been selected have 99999 in the order column.

Assume that all the fields you are interested in are selected by default. As no fields need to be added or deleted, you can press **F12** (Cancel) to return to the UTR Block Selection panel. The next section describes how to add and delete fields.

Replacing Fields

You may have noticed that the buffer pool block contains some fields that are in the general accounting data block (see Figure 57 on page 116). These fields are GETPAGES, SYN.READ, BUF.UPDT, and TOT.PREF.

In this example, the duplicated fields in the general accounting data block are replaced with class 3 suspensions and class 3 suspension times.

To modify the fields in the general accounting data block, enter a slash (/) in the *Modify* column next to General Accounting Data block and press **Enter**.

The UTR Field Selection panel is displayed.

```
 DGOBFLD3                  UTR Field Selection           Row 17 to 25 of 67

 Report set  . . . . . : Accounting Report
 Layout  . . . . . . . : SHORT
 Block label . . . . . : GENERAL

 1.  To include and order, modify/add a number in the Order column
 2.  To exclude a field, blank out the Order column

 Order  Label
        Description                                                 Length
        GETPAGES
        Average getpage requests for all buffer pools (QBACGET)          8
        BUF.UPDT
        Average buffer updates for all buffer pools (QBACSWS)            8
        SYN.READ
        Average synchronous reads for all buffer pools (QBACRIO)         8
        TOT.PREF
        Average all types of prefetch requests (ABCLSPR)                 8
 210    LOCK SUS
        Average all types of suspensions (ALTSUSP)                       8
 220    #LOCKOUT
        Sum of timeouts and deadlocks (ADTIMDLK)                         8
 99999  DESCRBE
        Average of DESCRIBE statements executed (QXDESC)                 7
 99999  ALL DML
        Average all SQL DML statements (ASCDML)                          7
 99999  #LOCKTB

Command ===> _________________________________________________________________
 F1=Help    F2=Split   F3=Exit    F5=Extend  F7=Up      F8=Down    F9=Swap
F10=Config F11=Filler F12=Cancel
```

Figure 65. Deleting Fields

Scroll to find the fields you do not want to report and blank out the numbers (170, 180, 190, and 200) in front of the fields. After you blank out the numbers, press **Enter** (to reorder the remaining fields) before scrolling further. Once you have blanked out all the unwanted fields, scroll to find the fields you want to add.

```
DGOBFLD3                  UTR Field Selection          Row 41 to 49 of 67

Report set  . . . . . : Accounting Report
Layout  . . . . . . . : SHORT
Block label . . . . . : GENERAL

1.  To include and order, modify/add a number in the Order column
2.  To exclude a field, blank out the Order column

Order  Label
       Description                                                  Length
  165  CL3 SUSP
       Average all Class 3 suspensions (ADTSUSC)                         8
99999  LOCK/LTCH TIME
       Avg time spent waiting due to lock latch suspensions(QWACAWTL)   14
99999  I/O SUSP.TIME
       Avg time spent waiting due to synch. I/O suspensions(QWACAWTI)   14
99999  OTH.READ TIME
       Avg time spent waiting due to read I/O performed (QWACAWTR)      14
99999  OTH.WRITE TIME
       Avg time spent waiting due to write I/O performed (QWACAWTW)     14
  168  CLS3 SUSP.TIME
       Avg time spent for all Class 3 suspensions (ADTSUST)             14
99999  #PARALL
       Sum parallel groups being executed (QXTOTGRP)                     7
99999  #PARUNS
       Sum parallel groups which fall back to seq. mode (ADTOTPFL)       7
99999  #PARRED

Command ===> ___________________________________________________________________
 F1=Help    F2=Split   F3=Exit    F5=Extend  F7=Up      F8=Down    F9=Swap
F10=Config F11=Filler F12=Cancel
```

Figure 66. Adding Fields

Enter any number between 160 and 210 in the *Order* column in front of the fields you want reported and press **Enter**. Scroll back to see that the newly inserted fields appear at the position you intended.

Note that you do not have to delete existing fields when you add new fields; you can insert new fields between existing fields or place them before or after the existing fields. Also, you can delete fields without replacing them with new fields. When the layout is generated the gaps are filled in.

There is no limitation to how many fields or blocks can be selected in a layout. However, you should carefully consider what information you really need to keep your reports from getting too long.

Displaying the Layout

Press **F3** (Exit) to return to the UTR Block Selection panel and press **F6** (Browse) to display the new layout. A sample report using the modified layout is displayed.

Note that, depending on the size of your screen, you might have to scroll right (**F11**) and down (**F8**) to see the entire report. Figure 67 on page 125 shows the right-hand side of the sample report.

```
 DGOFBRWS SYS94326.T153556.RA000.XXASP36.R0000110 --- Line 00000000 Col 047 126
********************************* Top of Data *********************************
DB2 PERFORMANCE MONITOR (V6)                                PAGE:
   ACCOUNTING REPORT - SHORT                      REQUESTED FROM:
                                                              TO:
                                                   INTERVAL FROM:
                                                              TO:

                          TOP NUMBER REQUESTED:


LECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME       CL3 SUSP LOCK
TCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME CLS3 SUSP.TIME #LOCK
----- ------- ------- ------- -------------- -------------- ------------- -----


              TYPE            SQLSTMT   CL7 CPU TIME  CL8 SUSP
               #OCCURS  CL7 ELAP.TIME  CL8 SUSP.TIME
------------  --------  -------------  -------------  --------






------------------------------------------------
ISTPREF  DYN.PREF  ASY.READ  BUF.UPDT  #SYNWRTE|
                                               |
------------------------------------------------
 Command ===> ________________________________________________ Scroll ===> CSR_
  F1=Help    F2=Split   F3=Exit    F5=Rfind   F7=Up      F8=Down    F9=Swap
 F10=Left   F11=Right  F12=Cancel
```

Figure 67. Browsing a Sample Layout

Saving the Layout

When you have completed the changes to your layout, exit from the UTR selection panels using **F3**. When you exit the UTR Block Selection panel, the UTR Save Layout Details panel is displayed, as shown in Figure 68. If you do not want to save the layout, use Cancel (**F12**) to leave the UTR Block Selection panel.

```
DGOBPSAV                 UTR Save Layout Details

Report set  . . . . . : Accounting Report

Provide the Layout Details and press Enter to save the layout
  or
Cancel to make further changes to the layout.

Layout  . . . BUFFER
Description   Modified short accounting report

Is each new record or entry to start on a new page ?  2  1=yes 2=no

 F1=Help    F12=Cancel
```

Figure 68. Saving the Layout

Although you are not recommended to do so, you can save the modified layout with the name of the layout you based it on. If you do this, bear in mind that the

DB2 PM documentation does not reflect your new layout. If you pressed **Enter**, you would save the modified layout with the name SHORT.

To keep the existing layout SHORT, and create an additional layout, give the layout a new name by typing over the existing name. This is the recommended approach. You can also type over the existing description.

On this panel you can also specify whether you want every entry on your 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.

In the above example we called the layout BUFFER and changed the description.

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 you just created is shown in the list of layouts. To exit from this panel, press **F3**.

Using the Layout

To use the new layout, make sure you specify the name of your DPMPARMS data set in the JCL. You can now run accounting reports using the new layout:

```
ACCOUNTING
     REPORT
        LAYOUT (BUFFER)
```

Specifying Time Zones

Read this section only if you want to modify the times used in reporting.

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 showing 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 IRF.

Time Zone Data Editor

Time zone information is entered using the time zone data editor which is accessed through the Interactive Report Facility (IRF).

To access the editor, select option 4 (*Maintain parameter data sets*) from the DB2 PM main menu. The Data Set Maintenance Menu is displayed.

```
DGOPMENU                   Data Set Maintenance Menu

Select one of the following.

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

Exception data set
'DGO.V6R1M0.THRESH'_________________________

DPMPARMS data set
'DGO.V6R1M0.DPMPARMS'_______________________






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

Figure 69. 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* on the screen.

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 OS/390 system under which the DB2 subsystem is running.

```
DGOPPLDS                Time Zone Data Editor              Row 1 to 1 of 1

Direction (1=East 2=West)
Action  Location          --   Time Zone  --   --   CPU Clock   --

                          Direction   HH   MM   Direction   HH   MM
____    ________________  _           __   __   _           __   __
****************************** Bottom of data ********************************










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

Figure 70. Time Zone Data Editor Panel

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 store clock 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 TIMEZONE 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

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.

```
DGOPPLDS                 Time Zone Data Editor               Row 1 to 2 of 2

Direction (1=East 2=West)
Action  Location           --   Time Zone  --   --   CPU Clock   --

                           Direction   HH   MM   Direction   HH   MM
____    USIBMSTOSQL1       2           08   00   1           00   00
____    *_______________   1           00   00   1           00   00
****************************** Bottom of data ********************************









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

Figure 71. 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 values have been specified in the *CPU Clock* column in order to indicate that the CPU clock is set to GMT. Note that 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 (*) has been specified for locations that do not have an individual entry. If no default is specified, no time zone adjustment is made to data from locations that do not have an entry.

Assume 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 DB2 PM 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 71.

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.

```
DGOPPLDS                 Time Zone Data Editor               Row 1 to 2 of 2

Direction (1=East 2=West)
Action  Location           --   Time Zone  --   --   CPU Clock   --

                           Direction   HH   MM   Direction   HH   MM
____    USIBMSTOSQL1       2           08   00   2           05   00
____    *______________    1           00   00   1           00   00
******************************* Bottom of data ********************************











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

Figure 72. Time Zone Data 2

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 DB2 PM 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 72.

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 one, USIBMSTOSQL2, in New York.

```
DGOPPLDS                  Time Zone Data Editor            Row 1 to 3 of 3

Direction (1=East 2=West)
Action  Location           --   Time Zone  --   --   CPU Clock   --

                           Direction   HH   MM   Direction   HH   MM
____    USIBMSTOSQL1       2           08   00   1           00   00
____    USIBMSTOSQL2       2           05   00   2           05   00
____    *                  1           00   00   1           00   00
****************************** Bottom of data ********************************

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

Figure 73. Time Zone Data 3

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.

Assume you again 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 DB2 PM 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 73.

As a result, data from all locations reflects San Francisco local time.

For more information about time zone processing, refer to the description of GLOBAL, and the appendixes of the *DB2 PM Report Reference*.

Correlation ID Translation

The correlation ID is a DB2 field that identifies the task executed by DB2. The correlation ID contains:

For batch jobs	Jobname
For TSO applications	Original authorization ID, that is, the logon user ID
For applications using the DB2 call attachment facility	Original authorization ID, that is, the logon user ID
For CICS transactions	Connection type, thread type, thread number, and the transaction ID
For IMS applications	PST number and PSBNAME of the application

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).

DB2 PM provides this function 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 3. The 12-Byte Correlation ID Field and the Default Translation

<table>
<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>
<tr><td>Batch</td><td colspan="8">Correlation name: job name</td><td colspan="4">Correlation number: blank</td></tr>
<tr><td>TSO, DB2 call attach</td><td colspan="8">Correlation name: original authorization ID</td><td colspan="4">Correlation number: blank</td></tr>
<tr><td>CICS</td><td colspan="4">Correlation number: pool thread</td><td colspan="4">Correlation name: transaction ID</td><td colspan="4"></td></tr>
<tr><td>IMS</td><td colspan="4">Correlation number: application PST</td><td colspan="8">Correlation name: application PSBNAME</td></tr>
<tr><td>RRS</td><td colspan="8">Correlation name: First 8 characters of the correlation ID as specified at RRS signon</td><td colspan="4">Correlation number: Last 4 characters of the correlation ID as specified at RRS signon</td></tr>
</table>

The correlation name and correlation number can be used as DB2 PM identifiers CORRNAME and CORRNMBR respectively in all report sets.

You can use CORRNAME and CORRNMBR to order data on reports, include and exclude input records, and present data on graphs.

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 to be used for a specific connection ID. Note that 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.

For further details, refer to the *DB2 PM Report Reference*.

If DB2 PM 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 which 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 DPMPARMS member MAINPACK. 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 DB2 PM main menu. The Data Set Maintenance Menu is displayed.

```
DGOPMENU                   Data Set Maintenance Menu

Select one of the following.

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

Exception data set
'DGO.V6R1M0.THRESH'_________________________

DPMPARMS data set
'DGO.V6R1M0.DPMPARMS'_______________________







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

Figure 74. Selecting Maintain MAINPACK Definitions

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* on the screen.

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.

```
DGOPPMDS           MAINPACK Definition Member Editor         ROW 1 TO 1 OF 1

Select one of the following MAINPACK definition codes for each
specified combination of Requesting Location, Connection ID and
Plan Name. Request EXIT when complete.

 Code      Description
  1        Package    ID of the first executed package
  2        Package    ID of the last executed package
  3        Collection ID of the first executed package
  4        Collection ID of the last executed package
  5        Location   of the first executed package
  6        Location   of the last executed package

Action  Requesting Location   Connection ID  Plan Name  Code
____    ________________      ________       ________    _
******************************* BOTTOM OF DATA ********************************




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

Figure 75. MAINPACK Definition Member Editor Panel

Enter the required information in the input fields. You can modify existing information by typing over it.

Action
Standard ISPF editor line commands. For example:
- **I** Generate a new line
- **D** Delete a line
- **R** Duplicate a line
- **M** Move a line.

Requesting Location
The 16-byte requesting location name. This name is used by DB2 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 in our example is a large plan consisting of several packages and it is used for many different kinds of transactions. Therefore we want to distinguish between the different executions of this plan.

The following figure shows the MAINPACK Definition Member Editor panel with the specifications.

```
DGOPPMDS           MAINPACK Definition Member Editor         Row 1 to 3 of 3

Select one of the following MAINPACK definition codes for each
specified combination of Requesting Location, Connection ID and
Plan Name. Request EXIT when complete.

 Code      Description
  1        Package    ID of the first executed package
  2        Package    ID of the last executed package
  3        Collection ID of the first executed package
  4        Collection ID of the last executed package
  5        Location   of the first executed package
  6        Location   of the last executed package

Action  Requesting Location  Connection ID  Plan Name  Code
____    *                    *              *          4
____    *                    *              DISTSERV   1
____    *                    *              CICSA      2
******************************* Bottom of data ********************************



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

Figure 76. Defining the MAINPACK

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 gives 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.

Using the MAINPACK and PACKAGE Identifiers (Example)

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 have been performed on behalf of a non-DB2 requester. One of them executes package CUSTINQR and the other one package ACCTCLOS. In this example, you only want to report activity performed by these non-DB2 requesters, that is, DISTSERV plans.

If you used PLANNAME to order the report, you could not distinguish between the two packages or the two transactions because they both execute plan DISTSERV. So to identify the transactions via 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 produced looks like this:

```
   LOCATION: VTAMA                                    DB2 PERFORMANCE MONITOR (V6)                                 PAGE: 1-1
      GROUP: N/P                                         ACCOUNTING REPORT - SHORT                      REQUESTED FROM: NOT SPECIFIED
     MEMBER: N/P                                                                                                   TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                             ORDER: MAINPACK                            INTERVAL FROM: 01/29/99 20:18:50.43
DB2 VERSION: V6                                               SCOPE: MEMBER                                        TO: 03/16/99 17:57:56.76

                            #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
MAINPACK                    #DISTRS #COMMIT FETCHES   OPENS  CLOSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
--------------------------- ------- ------- ------- ------- ------- ------- -------------- -------------- -------- -------- --------

CUSTINQR                          1       0    0.00    0.00    0.00    0.00       6.057603       5.969593    31.00    11.00     0.00
                                  0       1    3.00    3.00    3.00    0.00       0.095606       0.080949     0.00     0.00        0

    -------------------------------------------------------------------------------------------------------
    |PROGRAM NAME        TYPE     #OCCURS  SQLSTMT  CL7 ELAP.TIME   CL7 TCB TIME  CL8 SUSP.TIME  CL8 SUSP|
    |CUSTINQR            PACKAGE        1     3.00       5.560211       0.050515       1.167913      7.00|
    -------------------------------------------------------------------------------------------------------

ACCTCLOS                          1       0    0.00    0.00    0.00    0.00       0.271892       0.172424    17.00     2.00     0.00
                                  0       1    5.00    5.00    5.00    0.00       0.035721       0.014262     0.00     0.00        0

    -------------------------------------------------------------------------------------------------------
    |PROGRAM NAME        TYPE     #OCCURS  SQLSTMT  CL7 ELAP.TIME   CL7 TCB TIME  CL8 SUSP.TIME  CL8 SUSP|
    |ACCTCLOS            PACKAGE        1     3.00       0.078471       0.005308       0.006995      1.00|
    -------------------------------------------------------------------------------------------------------
```

Figure 77. 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 identification block.

Example 2: Using PACKAGE

Assume that during the reporting interval, used in this example, three transactions have been performed and that DB2 Accounting Trace classes 7 and 8 were 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 have been executed once within the transaction and the transactions have been 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                          DB2 PERFORMANCE MONITOR (V6)                             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: 01/29/99 20:18:50.43
DB2 VERSION: V6                                      SCOPE: MEMBER                                  TO: 03/16/99 17:57:56.76

PACKAGE: 'BLANK'.BOSNACOL.CUSTINQA

CUSTINQA           VALUE              CUSTINQA                  TIMES    CUSTINQA           AVERAGE TIME  AVG.EV  TIME/EVENT
------------------ ------------------ ------------------ ------------    ------------------ ------------  ------  ----------
TYPE               PACKAGE            ELAP-CL7 TIME-AVG      0.023464    LOCK/LATCH             0.000000    0.00         N/C
                                      CPU TIME               0.004540    SYNCHRONOUS I/O        0.012075    1.00    0.012075
LOCATION           'BLANK'              AGENT                0.004540    OTHER READ I/O         0.000000    0.00         N/C
COLLECTION ID      BOSNACOL             PAR.TASKS            0.000000    OTHER WRITE I/O        0.000000    0.00         N/C
PROGRAM NAME       CUSTINQA           SUSPENSION-CL8         0.012075    SERV.TASK SWITCH       0.000000    0.00         N/C
                                        AGENT                0.012075    ARCH.LOG(QUIESCE)      0.000000    0.00         N/C
OCCURRENCES                         2   PAR.TASKS            0.000000    ARCHIVE LOG READ       0.000000    0.00         N/C
SQL STMT - AVERAGE               3.00 NOT ACCOUNTED          0.211530    DRAIN LOCK             0.000000    0.00         N/C
SQL STMT - TOTAL                    6 AVG.DB2 ENTRY/EXIT         6.00    CLAIM RELEASE          0.000000    0.00         N/C
STOR PROC EXECUTED                 20 DB2 ENTRY/EXIT               12    PAGE LATCH             0.000000    0.00         N/C
UDF EXECUTED                        0                                    SCHED STORED PROC      0.000000    0.00         N/C
USED BY STOR PROC                   0 CPU SERVICE UNITS         11200    SCHEDULE UDF           0.000000    0.00         N/C
USED BY UDF                         0   AGENT                    8000    NOTIFY MESSAGES        0.000000    0.00         N/C
USED BY TRIGGER                     0   PAR.TASKS                3200    GLOBAL CONTENTION      0.000000    0.00         N/C
SUCC AUTH CHECK                     0                                    TOTAL CL8 SUSPENS.     0.012075    1.00    0.012075
```

Figure 78. Long Accounting Report Ordered by PACKAGE (Part 1 of 3)

```
   LOCATION: VTAMA                             DB2 PERFORMANCE MONITOR (V6)                                    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: 01/29/99 20:18:50.43
DB2 VERSION: V6                                         SCOPE: MEMBER                                         TO: 03/16/99 17:57:56.76

PACKAGE: 'BLANK'.BOSNACOL.CUSTINQB

CUSTINQB            VALUE                 CUSTINQB                    TIMES    CUSTINQB            AVERAGE TIME  AVG.EV  TIME/EVENT
------------------  ------------------    ------------------  ------------    ------------------  ------------  ------ -----------
TYPE                PACKAGE               ELAP-CL7 TIME-AVG       0.092622    LOCK/LATCH              0.000000    0.00         N/C
                                          CPU TIME                0.004247    SYNCHRONOUS I/O         0.027006    1.00    0.027006
LOCATION            'BLANK'                 AGENT                 0.004247    OTHER READ I/O          0.000000    0.00         N/C
COLLECTION ID       BOSNACOL                PAR.TASKS             0.000000    OTHER WRITE I/O         0.000000    0.00         N/C
PROGRAM NAME        CUSTINQB              SUSPENSION-CL8          0.027006    SERV.TASK SWITCH        0.000000    0.00         N/C
                                            AGENT                 0.019006    ARCH.LOG(QUIESCE)       0.000000    0.00         N/C
OCCURRENCES                          2      PAR.TASKS             0.008000    ARCHIVE LOG READ        0.000000    0.00         N/C
SQL STMT - AVERAGE                3.00    NOT ACCOUNTED           0.258774    DRAIN LOCK              0.000000    0.00         N/C
SQL STMT - TOTAL                     6    AVG.DB2 ENTRY/EXIT          6.00    CLAIM RELEASE           0.000000    0.00         N/C
STOR PROC EXECUTED                   0    DB2 ENTRY/EXIT                12    PAGE LATCH              0.000000    0.00         N/C
UDF EXECUTED                         0                                        SCHED STORED PROC       0.000000    0.00         N/C
USED BY STOR PROC                    0    CPU SERVICE UNITS          11920    SCHEDULE UDF            0.000000    0.00         N/C
USED BY UDF                          0      AGENT                     7100    NOTIFY MESSAGES         0.000000    0.00         N/C
USED BY TRIGGER                      0      PAR.TASKS                 4820    GLOBAL CONTENTION       0.000000    0.00         N/C
SUCC AUTH CHECK                      0                                        TOTAL CL8 SUSPENS.      0.027006    1.00    0.027006
```

Figure 78. Long Accounting Report Ordered by PACKAGE (Part 2 of 3)

```
   LOCATION: VTAMA                             DB2 PERFORMANCE MONITOR (V6)                                    PAGE: 1-3
      GROUP: N/P                                  ACCOUNTING REPORT - LONG                        REQUESTED FROM: NOT SPECIFIED
     MEMBER: N/P                                                                                              TO: NOT SPECIFIED
  SUBSYSTEM: SSDQ                                       ORDER: PACKAGE                             INTERVAL FROM: 01/29/99 20:18:50.43
DB2 VERSION: V6                                         SCOPE: MEMBER                                         TO: 03/16/99 17:57:56.76

PACKAGE: 'BLANK'.'BLANK'.CUSTINQC

CUSTINQC            VALUE                 CUSTINQC                    TIMES    CUSTINQC            AVERAGE TIME  AVG.EV  TIME/EVENT
------------------  ------------------    ------------------  ------------    ------------------  ------------  ------ -----------
TYPE                DBRM                  ELAP-CL7 TIME-AVG       0.031617    LOCK/LATCH              0.000000    0.00         N/C
                                          CPU TIME                0.004170    SYNCHRONOUS I/O         0.027447    1.00    0.027447
LOCATION            'BLANK'                 AGENT                 0.004170    OTHER READ I/O          0.000000    0.00         N/C
COLLECTION ID       'BLANK'                 PAR.TASKS             0.000000    OTHER WRITE I/O         0.000000    0.00         N/C
PROGRAM NAME        CUSTINQC              SUSPENSION-CL8          0.027447    SERV.TASK SWITCH        0.000000    0.00         N/C
                                            AGENT                 0.019000    ARCH.LOG(QUIESCE)       0.000000    0.00         N/C
OCCURRENCES                          1      PAR.TASKS             0.008447    ARCHIVE LOG READ        0.000000    0.00         N/C
SQL STMT - AVERAGE                3.00    NOT ACCOUNTED           0.268774    DRAIN LOCK              0.000000    0.00         N/C
SQL STMT - TOTAL                     3    AVG.DB2 ENTRY/EXIT          6.00    CLAIM RELEASE           0.000000    0.00         N/C
STOR PROC EXECUTED                  21    DB2 ENTRY/EXIT                 6    PAGE LATCH              0.000000    0.00         N/C
UDF EXECUTED                         0                                        SCHED STORED PROC       0.000000    0.00         N/C
USED BY STOR PROC                    0    CPU SERVICE UNITS          12400    SCHEDULE UDF            0.000000    0.00         N/C
USED BY UDF                          0      AGENT                     8600    NOTIFY MESSAGES         0.000000    0.00         N/C
USED BY TRIGGER                      0      PAR.TASKS                 3800    GLOBAL CONTENTION       0.000000    0.00         N/C
SUCC AUTH CHECK                      0                                        TOTAL CL8 SUSPENSIONS   0.27447     1.00    0.027447
                                                                                                          1
```

Figure 78. Long Accounting Report Ordered by PACKAGE (Part 3 of 3)

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.

Note that in the accounting long report example shown in Figure 78 each entry begins on a new page, even though all the entries fit on a single page. To print all the 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. All you need to do is press F3 on the UTR Block Selection panel to display the UTR Save Layout Details panel (see Figure 68 on page 125). 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 Descriptions

You can change the description of exception fields as they are printed for the exception records in batch reports. To change the description, create a member called EXCHANGE in your DPMPARMS data set using the ISPF editor. For each field that requires a change in the description, enter a line into this member that contains the field name and the new description, separated by a comma without spaces.

Chapter 11. Reporting Distributed Data

Read this section only if you want to report distributed data.

DB2 supports two methods of remote access between a requesting relational database management system (DBMS) and a serving DBMS. The two types of access are DB2 private protocol and DRDA(R) protocol. DB2 chooses between the two access methods based on the SQL statements contained in the application process.

In *DB2 private protocol* DB2 uses communication connections that are specific to DB2, so only DB2 subsystems can communicate using this connection.

In *DRDA protocol* an application can explicitly connect with another relational DBMS that supports the Distributed Relational Database Architecture(TM) (DRDA). The application is not restricted to accessing data only at DB2 subsystems.

Each DB2 subsystem has an additional address space whenever the Distributed Data Facility (DDF) of DB2 has been started. It is referred to as the DDF address space. The DDF address space uses Virtual Telecommunication Access Method (VTAM), or TCP/IP to communicate with other database management systems.

DB2 PM reports the following DB2 distributed data:

- The **statistics** report set shows:
  - DDF address space CPU times
  - Statistics for each DB2 remote location for DB2 private protocol
  - Aggregate statistics for all locations for DRDA protocol
  - Other, nonlocation-specific DDF information.
- The **accounting** report set shows information for specificthreads participating in distributed activity. Information is shown both for 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.
- The **SQL activity** report set shows all SQL data within threads.The SQL statements are reported at the location where they were executed, that is, both at the requester and the server.
- The **explain** report set can show information for packages bound at a remote location.

  If a list of plans to be explained contains a remotely bound package, DB2 PM EXPLAIN automatically connects to the server and explains the remote package. Alternatively, you can specify the server location to which DB2 PM EXPLAIN is to connect and the plans and packages you want explained.

Terminology

The following terms are used by DB2 PM to refer to distributed activity:

Distributed data facility (DDF) data
: DB2 instrumentation data showing distributed activity.

Local (reporting) location
: The location that is the focus of the report (the location indicated in the report header) is considered the *local* or *reporting* location.

Distributed Data

Location names and other information about connections with remote systems are held in the Communications Database. For more information refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide*.

Remote location
All other locations requesting or providing data for the local location.

Requester location
The name of the location that requests data from another location.

Server location
The name of the location that performs work on behalf of a requester location.

Thread
The DB2 structure that describes an application's connection, traces its progress, provides resource function processing capability, and delimits its accessibility to DB2 resources and services. Most DB2 functions execute under a thread structure. DB2 PM uses the following categorization of DB2 threads:

Allied thread
A thread that does not involve distributed activity.

Allied-distributed thread
A thread that is initiated by a DB2 attach such as TSO, IMS, or CICS and that requests data from one or more server locations.

For example, when DB2 private protocol is used, an application's SQL statements can refer to a three-part table name or an alias residing at a remote location. When DRDA protocol is used, the statements can refer to a package that has previously been bound at a remote location.

Database access thread (DBAT)
A thread initiated, created, and performing work on behalf of a remote (requester) location.

In the case of SQL activity, all the SQL statements of the DBAT refer to local objects. When DB2 private protocol is used, the statements are dynamically bound on behalf of the requester location, and when DRDA protocol is used, they are part of a package.

Database access thread — distributed (DBAT-distributed)
A thread initiated by a requester location to a server location that in turn requests data from another server location.

The SQL statements of a DBAT-distributed thread are part of a package bound for DRDA protocol but they refer to a three-part table name or an alias residing at a remote location.

Refer to Figure 79 for an example of the different thread types.

Nondistributed transaction
DB2 activity initiated 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 an allied thread is reused, a nondistributed transaction is DB2 activity between two signon (resignon) points.

Distributed transaction

DB2 activity initiated at one (requester) location and performed at one or more remote (server) locations.

Distributed transactions consist of local activity represented by an allied-distributed thread, and remote activity represented by one or more DBATs. Therefore, for a full picture of a distributed transaction, instrumentation data for the allied-distributed thread as well as all the corresponding DBATs is needed.

In the following figures, the beginning and the end of a thread is marked with =. Threads are drawn with solid lines and distributed calls with arrows.

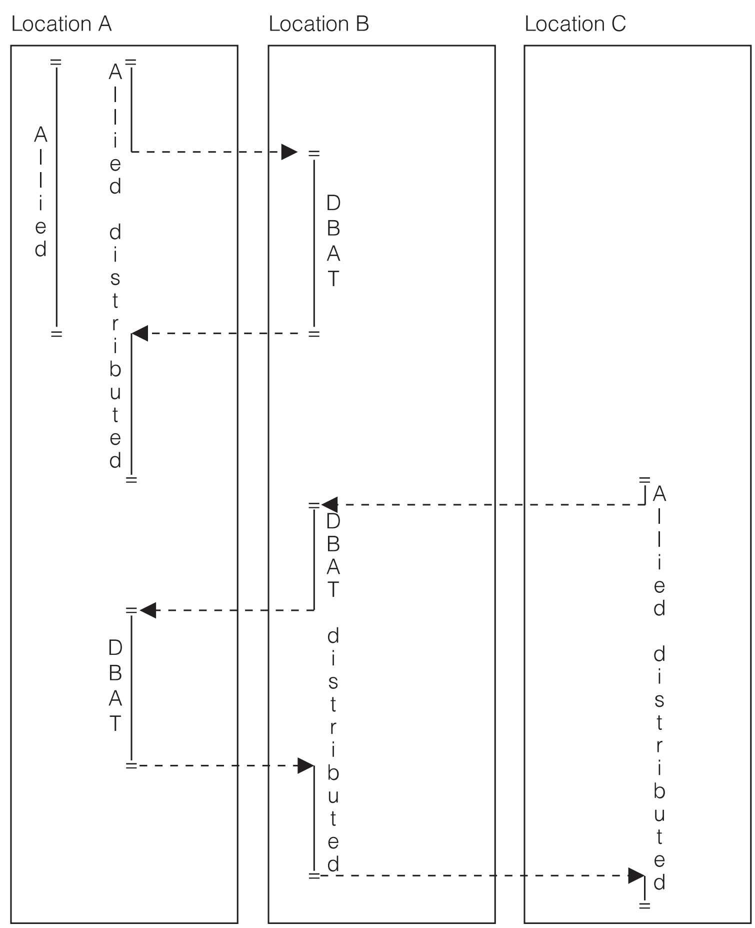

Figure 79. Thread Types

Figure 79 shows an allied thread, an allied-distributed thread, and a DBAT at location A. A DBAT at location B is performing work on behalf of location A, and a DBAT-distributed thread at location B is requesting work from location A on behalf of location C. An allied-distributed thread is shown at location C.

Distributed Processing

DB2 PM can produce reports for a single DB2 location or for a number of different DB2 locations. Input data sets from several locations can be concatenated and processed in DB2 PM.

Reports and Traces

You can produce reports and traces that include data from one or several locations. When a report or trace contains information from more than one location, the data is reported separately for each subsystem and is ordered by location in alphabetic order. If distributed activity takes place, the distributed call is reported, but the work performed at the server location is not reported.

The following is reported for every location:

- Nondistributed transactions, that is, the allied threads at the reporting location.
- Local activity of distributed transactions originating at the reporting location, that is, 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, that is, the DBATs at the reporting location.

Input Data

DB2 PM reports DB2 distributed processing with instrumentation data originating at a single DB2 location or at a number of different DB2 locations. Multiple input data sets can be logically concatenated in the DD statements for INPUTDD and processed in a single DB2 PM run. You can specify any combination of SMF, GTF, DPMOUT, or Online Monitor report data sets. The data is grouped by location and reported according to the specified DB2 PM commands.

Selecting Threads

You can select only certain threads or thread types to be reported using the INCLUDE/EXCLUDE option.

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 INCLUDE/EXCLUDE filters on REQLOC to include only data for that location.

If you want to report only distributed data, use EXCLUDE THREADTYPE(ALLIED) to exclude data for allied threads.

Examples

The following examples describe different kinds of distributed transactions and show which threads are included in reports and traces.

Distributed Transaction between DB2 for MVS Systems

In Figure 80 all three locations are DB2 systems.

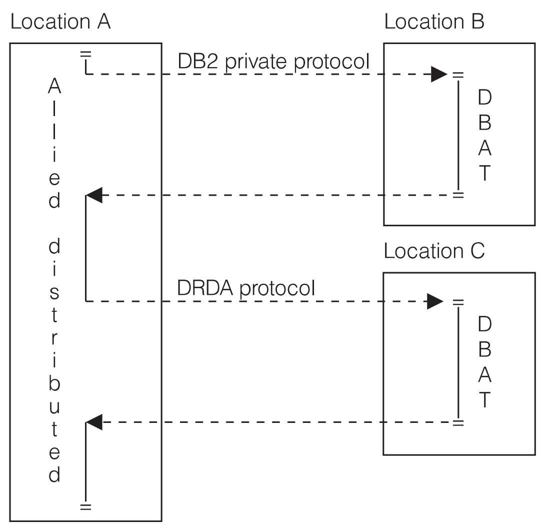

Figure 80. Distributed Transaction between DB2 Systems

A distributed transaction originates at location A. Apart from some local DB2 activity, it accesses data at location B using DB2 private protocol and data at location C using DRDA protocol.

This transaction consists of an allied-distributed thread at location A and one DBAT for each server at location B and C.

Table 4. Example 1: DB2 to DB2 Transaction

Location A	Location B	Location C
Allied-distributed thread	—	—
—	DBAT (DB2 private protocol)	DBAT (DRDA protocol)

Merged reports for locations B and C do not show any data because no threads are initiated at these locations.

Distributed Transaction between DB2/MVS and Non-DB2/MVS Systems

In Figure 81 on page 146 one of the locations is a DB2 system (location A) and two are non-DB2 systems (Rdb_name_1 and Rdb_name_2).

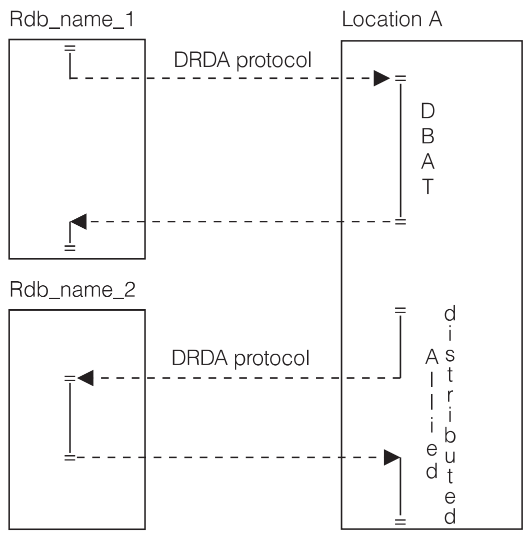

Figure 81. Distributed Transaction between DB2 and Non-DB2 Systems

One distributed transaction originates at Rdb_name_1 accessing data at location A and another one originates at location A accessing data at Rdb_name_2. Because the activity takes place between non-DB2 systems and a DB2 system, DRDA protocol is used.

The first transaction consists of a DBAT at location A and the second of an allied-distributed thread at location A.

The following threads are shown:

Table 5. Example 2: DB2 and Non-DB2 Transaction

Rdb_name_1	Rdb_name_2	Location A
—	—	DBAT and allied-distributed thread

The activity performed at non-DB2 systems cannot be reported by DB2 PM.

Distributed Transaction Involving a DBAT-Distributed Thread

In Figure 82 all three locations are DB2 systems.

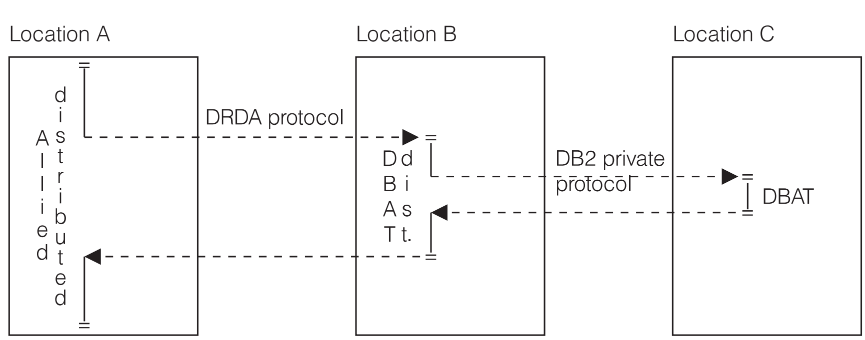

Figure 82. Distributed Transaction Involving a DBAT-Distributed Thread

A distributed transaction originates at location A. Apart from some local DB2 activity, it accesses data at location B using DRDA protocol and, in the same unit of work, data at location C using DB2 private protocol.

This transaction consists of an allied-distributed thread at location A, a DBAT-distributed thread at location B, and a DBAT at location C.

The following threads are shown:

Table 6. Example 3: Reports Involving a DBAT-Distributed Thread

Location A	Location B	Location C
Allied-distributed thread	—	—
—	DBAT-distributed thread	—
—	—	DBAT at location C

Chapter 12. Reporting Data Sharing Information

Read this section only if you want to monitor the performance of members of data sharing groups.

Data sharing gives individual DB2 subsystems full access to databases that are on shared DASD. The DB2 subsystems sharing the data belong to a *data sharing group* and each subsystem is considered a *member* of the group.

In a data sharing environment, you need to be able to monitor the performance of entire data sharing groups as well as individual members of a group. You can do this by generating reports or traces that combine performance information for all the members, called *group-scope reports*, or by generating ordinary DB2 PM reports for the individual members.

Group-scope reports are available in the accounting, locking, audit, and statistics report sets.

All DB2 PM 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 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.

IRF graphs provide a visual means of investigating trends for a group. You can produce, for example, a graph summarizing locking activity for an entire group, or a graph showing how the different members have used group buffer pools.

Monitoring Individual Members

You can monitor various aspects of performance for individual members of a group using any of the DB2 PM report sets.

Example of a Member-Scope Locking Report

Lockout 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 in that holders and waiters of lockup are shown regardless on which they execute. Member scope and group scope only influence the summarization of the report.

To generate the report, specify the following:

```
⋮
LOCKING
     REPORT
        LEVEL (LOCKOUT)
        ORDER(DATABASE-PAGESET)
⋮
```

Member-scope report is the default, so you do not have to specify the SCOPE option. Note that you must specify the ORDER option if you want to order the report by page set within a database.

The following example shows a member-scope lockout report for group DSHGRPXX, which has two members, FIRST and SECOND. The report is two pages long because a new page is started when the member being reported changes.

The first page of the report shows the locking activity of the threads that have executed in member FIRST.

```
   LOCATION: SYDNEY                               DB2 PERFORMANCE MONITOR (V6)                      PAGE: 1-1
      GROUP: DSHGRPXX                                                                     REQUESTED FROM: NOT SPECIFIED
     MEMBER: FIRST                                 LOCKING REPORT - LOCKOUT                             TO: NOT SPECIFIED
  SUBSYSTEM: DB22                                   ORDER: DATABASE-PAGESET                    ACTUAL FROM: 01/19/99 12:15:00.21
DB2 VERSION: V6                                           SCOPE: MEMBER                                 TO: 01/19/99 13:27:56.09

DATABASE           --- L O C K   R E S O U R C E ---                     --------------- A G E N T S ----------------  BLOCKER/
 PAGESET           TYPE      NAME                   TIMEOUTS DEADLOCKS  MEMBER   PLANNAME CONNECT  CORRNAME CORRNMBR  HOLDER WAITER
------------------ --------- ---------------------- -------- ---------  -------- -------- -------- -------- --------  ------ ------
DBASE9             ROW       PAGE=X'000021'                0         3  FIRST    D3APP01  BATCH    RUNPR01  'BLANK'        2      1
 TSPACEXX                    ROW =X'03'                                 SECOND   D3APPBB  BATCH    RUNPRBB  'BLANK'        1      1
                                                                        SECOND   D3APPDD  BATCH    RUNPRDD  'BLANK'        0      2
                   INDEXPAGE PAGE=X'002393'                0         1  FIRST    D3APP02  BATCH    RUNPR02  'BLANK'        0      1
                             SUBP=X'01'

                   ** LOCKOUTS FOR TSPACEXX      **        0         4

** TOTAL - DBASE9 **                                       0         4

DBASE10
 TSPACEZZ          DATAPAGE  PAGE=X'000055'                0         3  FIRST    D3APP03  BATCH    RUNPR03   'BLANK'       2      1
                                                                        FIRST    D3APP05  BATCH    RUNPR05   'BLANK'       0      3
                   ** LOCKOUTS FOR TSPACEZZ      **        0         3

** GRAND TOTAL **                                          0         7
```

Figure 83. Member-Scope Locking Lockout Report, Page 1

The second page of the report shows the locking activity of the threads that have executed in member SECOND.

```
   LOCATION: SYDNEY                                  DB2 PERFORMANCE MONITOR (V6)                        PAGE: 2-1
      GROUP: DSHGRPXX                                                                       REQUESTED FROM: NOT SPECIFIED
     MEMBER: SECOND                                   LOCKING REPORT - LOCKOUT                          TO: NOT SPECIFIED
  SUBSYSTEM: DB22                                      ORDER: DATABASE-PAGESET                 ACTUAL FROM: 01/19/99 12:15:00.21
DB2 VERSION: V6                                              SCOPE: MEMBER                              TO: 01/19/99 13:27:56.09

DATABASE           --- L O C K   R E S O U R C E ---                          --------------- A G E N T S ----------------
 PAGESET           TYPE      NAME                       TIMEOUTS DEADLOCKS MEMBER   PLANNAME CONNECT  CORRNAME CORRNMBR  HOLDER WAITER
------------------ --------- ----------------------- -------- --------- -------- -------- -------- -------- --------  ------ ------
DBASE9             ROW       PAGE=X'000021'                 0         4 FIRST    D3APP01  BATCH    RUNPR01  'BLANK'        2      1
 TSPACEXX                    ROW =X'03'                                FIRST    D3APP02  BATCH    RUNPR02  'BLANK'        1      1
                                                                       SECOND   D3APPAA  BATCH    RUNPRAA  'BLANK'        0      2
                                                                       SECOND   D3APPBB  BATCH    RUNPRBB  'BLANK'        1      1
                   ** LOCKOUTS FOR TSPACEZZ       **        0         4

DBASE10            ROW       PAGE=X'000021'                 0         4 FIRST    D3APP01  BATCH    RUNPR01  'BLANK'        1      0
 TSPACEZZ                    ROW =X'03'                                FIRST    D3APP02  BATCH    RUNPR02  'BLANK'        0      1
                                                                       SECOND   D3APPCC  BATCH    RUNPRCC  'BLANK'        2      1
                                                                       SECOND   D3APPEE  BATCH    RUNPREE  'BLANK'        0      2
                   INDEXPAGE PAGE=X'000033'                 0         5 SECOND   D3APPBB  BATCH    RUNPRBB  'BLANK'        1      1
                             SUBP=X'03'                                SECOND   D3APPCC  BATCH    RUNPRCC  'BLANK'        1      0
                                                                       SECOND   D3APPDD  BATCH    RUNPRDD  'BLANK'        0      2
                   ** LOCKOUTS FOR TSPACEZZ       **        0         5

** GRAND TOTAL **                                           0         9
```

Figure 84. Member-Scope Locking Lockout Report, Page 2

Monitoring Entire Groups

Use group-scope reports to get 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 accounting group-scope reports merge instrumentation data produced by the individual group members and present it for the entire group.
- The locking group-scope reports give a full picture of the locking activity within the entire data sharing group.
- The statistics group-scope 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 audit group-scope reports give a comprehensive view of the access to shared resources by the users of the members of group. If you want, for example, a summary of users belonging to various members of a group who accessed, or attempted to access, page sets on shared databases, you could 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 lockout report, you need to specify the following:

```
⋮

LOCKING
      REPORT
         LEVEL (LOCKOUT)
         SCOPE (GROUP)

⋮
```

Note that the default order of a group scope report is DATABASE-PAGESET.

Data Sharing

The following example shows a group-scope lockout 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 for the database, then the page set, and lastly for the individual member.

```
   LOCATION: SYDNEY                                DB2 PERFORMANCE MONITOR (V6)                        PAGE: 1-1
      GROUP: DSHGRPXX                                                                        REQUESTED FROM: NOT SPECIFIED
                                                    LOCKING REPORT - LOCKOUT                             TO: NOT SPECIFIED
                                                     ORDER: DATABASE-PAGESET                    ACTUAL FROM: 01/19/99 12:15:00.21
DB2 VERSION: V6                                            SCOPE: GROUP                                  TO: 01/19/99 13:27:56.09

DATABASE
 PAGESET           --- L O C K   R E S O U R C E ---                       --------------- A G E N T S ----------------
  MEMBER           TYPE      NAME                       TIMEOUTS DEADLOCKS  MEMBER   PLANNAME CONNECT  CORRNAME CORRNMBR  HOLDER WAITER
------------------ --------- ----------------------- -------- ---------  -------- -------- -------- -------- --------  ------ ------
DBASE9
 TSPACEXX
  FIRST            ROW       PAGE=X'000021'                 0         3  FIRST    D3APP01  BATCH    RUNPR01  'BLANK'        2      1
                             ROW =X'03'                                  SECOND   D3APPBB  BATCH    RUNPRBB  'BLANK'        1      1
                                                                         SECOND   D3APPDD  BATCH    RUNPRDD  'BLANK'        0      2
                   INDEXPAGE PAGE=X'002393'                 0         1  FIRST    D3APP02  BATCH    RUNPR02  'BLANK'        0      1
                             SUBP=X'01'

                   ** LOCKOUTS FOR FIRST          **        0         4

  SECOND           ROW       PAGE=X'000021'                 0         4  FIRST    D3APP01  BATCH    RUNPR01  'BLANK'        2      1
                             ROW =X'03'                                  FIRST    D3APP02  BATCH    RUNPR02  'BLANK'        1      1
                                                                         SECOND   D3APPAA  BATCH    RUNPRAA  'BLANK'        0      2
                                                                         SECOND   D3APPBB  BATCH    RUNPRBB  'BLANK'        1      1

                   ** LOCKOUTS FOR SECOND         **        0         4

 ** GROUP TOTAL **                                          0         8


DBASE10
 TSPACEZZ
  FIRST            DATAPAGE  PAGE=X'000055'                 0         3  FIRST    D3APP03  BATCH    RUNPR03  'BLANK'        2      1
                                                                         FIRST    D3APP05  BATCH    RUNPR05  'BLANK'        0      3
                   ** LOCKOUTS FOR FIRST          **        0         3

  SECOND           ROW       PAGE=X'000021'                 0         4  FIRST    D3APP01  BATCH    RUNPR01  'BLANK'        1      0
                             ROW =X'03'                                  FIRST    D3APP02  BATCH    RUNPR02  'BLANK'        0      1
                                                                         SECOND   D3APPCC  BATCH    RUNPRCC  'BLANK'        2      1
                                                                         SECOND   D3APPEE  BATCH    RUNPREE  'BLANK'        0      2
                   INDEXPAGE PAGE=X'000033'                 0         5  SECOND   D3APPBB  BATCH    RUNPRBB  'BLANK'        1      1
                             SUBP=X'03'                                  SECOND   D3APPCC  BATCH    RUNPRCC  'BLANK'        1      0
                                                                         SECOND   D3APPDD  BATCH    RUNPRDD  'BLANK'        0      2
                   ** LOCKOUTS FOR SECOND         **        0         5

 ** GROUP TOTAL **                                          0         8

** GRAND TOTAL **                                           0        16
```

Figure 85. Group-Scope Locking Lockout Report for DSHGRPXX

Note that a group total is printed for the entire group when the database being monitored changes. The grand total shows the timeouts and deadlocks in all databases for the entire group.

Group-Scope Statistics

The statistics group-scope reports show three categories of information summarized on group level:

Highlights
: They present values such as the total number of threads and commitments for the entire group.

Data sharing locks
: They present locking information for shared resources for all members.

Buffer pool data
: They present 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 DB2 PM identifiers you specified and 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.

Graphs

You can produce graphs for members of a data sharing group as well as for the entire group. For more information about graphs in general, refer to “Chapter 13. Producing Graphs” on page 157.

Producing a Graph Showing Members

This example shows how to produce a graph showing the group buffer pool usage by individual members of a data sharing group.

```
DGOGACDI          Graphics - Accounting by DB2 PM Identifier
Command ===> ___________________________________________________________

Update the following fields, then press Enter.

Graph title . . . Group BPool Usage per Member
SAVE data set . . ACCTSAVE
Location  . . . . SYDNEY
Group . . . . . . DSHGRPA
Field name  . . . QAGASW    +
Qualifier . . . . TOTAL_________________________________________________ +

                        YY  MM  DD  HH  MM  SS  TH
From date/time  . . . . 93  04  01  15  00  00  00
To date/time  . . . . . 93  04  01  16  00  00  00

X-Axis divisions
1    Number of units per division
2    Units (1=Days 2=Intervals)

DB2 PM Identifiers          Values
MEMB  +                     FIRST             SECOND
                            THIRD             FOURTH
____  +                     ________________  ________________
                            ________________  ________________
____  +                     ________________  ________________
                            ________________  ________________
```

Figure 86. Accounting by DB2 PM Identifier Entry Screen

The requested graph shows the average number of changed pages written to all the group buffer pools for each 15-minute interval (specified at SAVE data set generation time) between 15:00 and 16:00 on 01 April 1993. The average number of changed pages is plotted individually for the four members of the data sharing group. This way you can compare how the group buffer pool is used by each member.

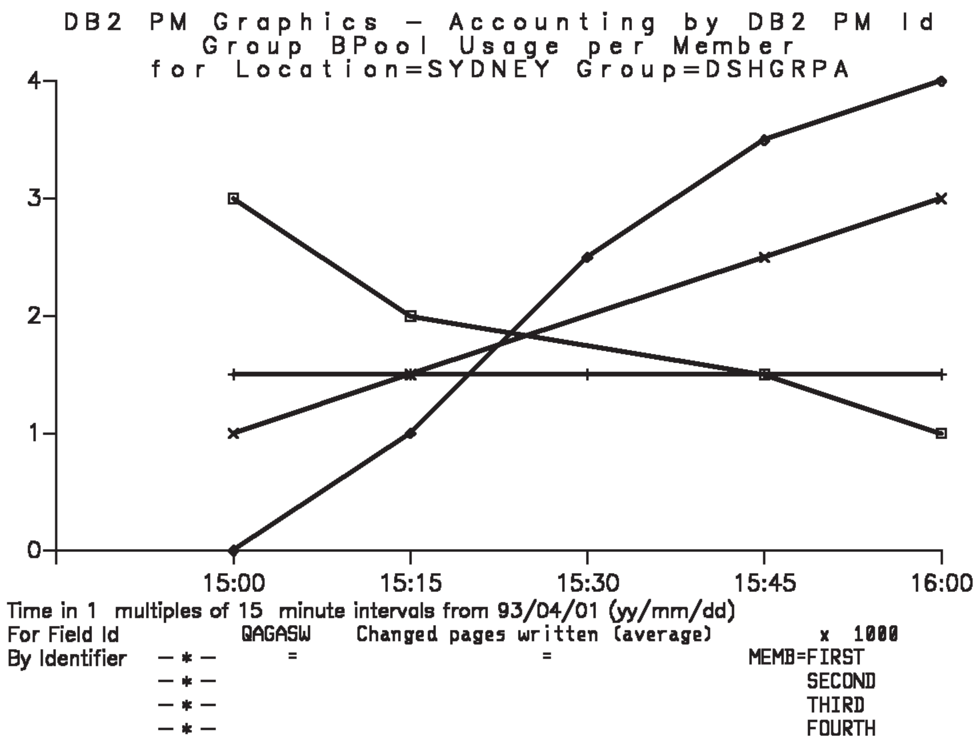

Figure 87. Accounting by DB2 PM Identifier Graph

Producing a Graph Showing an Entire Group

This example shows how to produce a graph showing the lock suspensions on a page set of a shared database.

```
DGOGDIST              Graphics - Frequency Distribution
Command ===> ____________________________________________________________

Update the following fields, then press Enter.

Graph title . . . Normal Resume Time per Group
Data set  . . . . 'XXASP24.LOCKING.DIST'
Location  . . . . SYDNEY
Group . . . . . . DSHGRPA
Field name  . . . LNRMLRET  +
Qualifier . . . . _______________________________________________________

                        YY  MM  DD  HH  MM  SS  TH
From date/time  . . . . 94  06  27  00  00  00  00
To date/time  . . . . . 94  06  27  23  59  59  99

DB2 PM Identifiers          Values
MEMB  +                     *
DTBS  +                     DBASE01
PSET  +                     PSET05
```

Figure 88. Frequency Distribution Entry Screen

The requested graph shows the frequency distribution of suspensions that resumed normally on page set PSET05 and DBASE01 and the elapsed time of the suspensions. The suspensions were monitored on 27 June 1994, on all the members of data sharing group DSHGRPA.

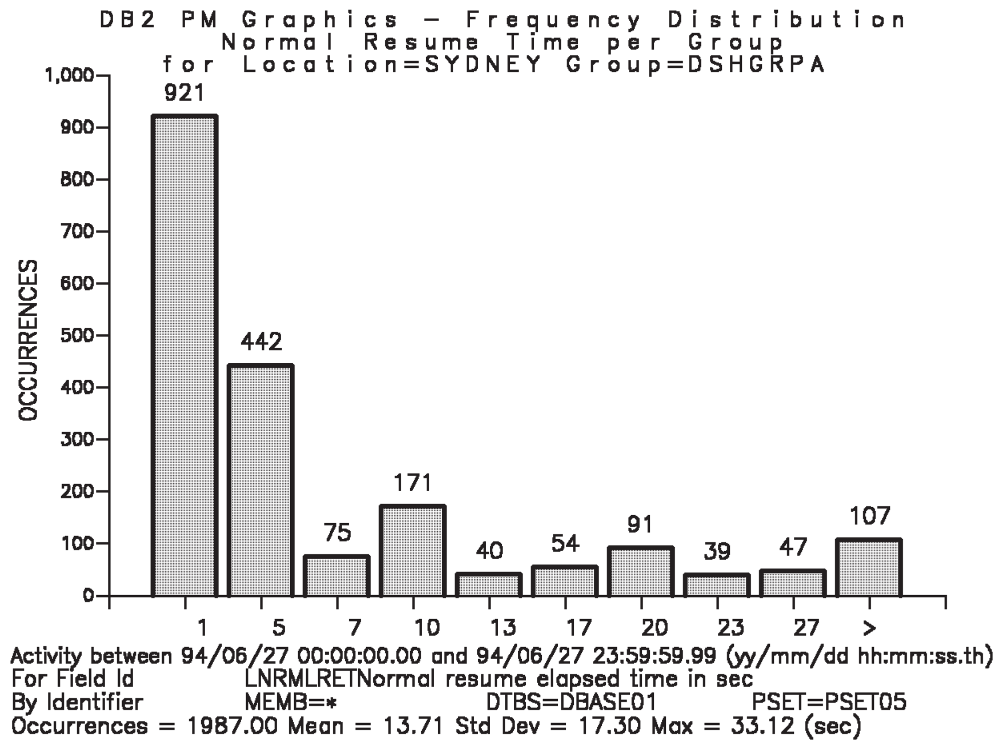

Figure 89. Frequency Distribution Graph

The graph shows the number suspensions on the shared database grouped by suspension time. It shows that a large part of the suspensions, 921, resumed within a second. 442 suspensions lasted between 1 and 5 seconds and 107 suspensions lasted over 27 seconds.

Collecting Input Data for Group-Scope Reports

When you want to produce group-scope reports, you need to obtain the instrumentation data for all the members of a group.

For regular monitoring, you most likely collect the performance data into SMF data sets. If all the 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 are running on different OS/390 systems, you need to concatenate the SMF data sets from all the OS/390 systems when you specify the input to DB2 PM.

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 them to generate group-scope reports. For more information about collecting instrumentation data, refer to “Chapter 5. Collecting Performance Data” on page 27.

Chapter 13. Producing Graphs

Use the IRF graphics for both short-term and long-term analysis of DB2 performance. The graphs are especially useful in determining trends in DB2 performance.

You can produce the graphs from saved accounting and statistics data or from frequency distribution data. You can either view the graphs online or print them.

To produce graphs, access the IRF and select option 2 *Display and print graphs* from the DB2 PM main menu, as shown in Figure 90.

```
DGOFMENU            IBM Database 2 Performance Monitor

Select one of the following.

2_ 1. Create and execute DB2 PM commands
   2. Display and print graphs
   3. View online DB2 activity
   4. Maintain parameter data sets
   5. Customize DB2 PM report and trace layouts
   6. Exception profiling
```

Figure 90. DB2 PM Main Menu

Next select the type of graph you want to plot.

```
DGOGMENU                 Graphics Selection Menu

Select one of the following.

   1. Accounting by field identifiers
   2. Accounting by DB2 PM identifiers
   3. Statistics
   4. Frequency distribution
```

Figure 91. Graphics Selection Menu

The following types of graphs are available:

- *Accounting by field identifier*

  Use the accounting by field identifier graphs to compare the values of up to four accounting fields.

  For example, you could produce an accounting field identifier graph showing the elapsed, CPU, and suspension times for a particular plan.
- *Accounting by DB2 PM identifier*

  Use the accounting by DB2 PM identifier graphs to compare combinations of DB2 PM identifiers (such as plans or users) against one accounting field. You could, for example, compare the elapsed application times of different plans.
- *Statistics*

  Use the statistics graphs to compare the values of up to four statistics fields.

  For example, you could produce a graph comparing the different CPU times for a DB2 subsystem during a statistics interval.

There are no statistics by DB2 PM identifier graphs, because in statistics all events represent the entire system.

- *Frequency distribution*

  Use frequency distribution graphs to analyze the frequency at which a field is between certain values.

  Frequency distribution graphs plot data generated by the DB2 PM DISTRIBUTE command. These graphs can be generated for selected fields from accounting, I/O, locking, SQL activity, and utility activity data.

  You can, for example, produce a graph showing different ranges of elapsed times to determine whether a large part of a group of plans have performance problems.

On the graph specification panel, specify the input data set (an accounting or statistics save data set or a DISTRIBUTE data set) and other graph information. After you have completed the specifications, press **Enter** to produce the graph.

```
DGOGACFI        Graphics - Accounting by Field Identifiers

Update the following fields, then press Enter.

Graph title . . . . . . Example Accounting Graph______
SAVE data set . . . . . ACSAVDD___________________________________________
Location  . . . . . . . HODB2P______________
Group . . . . . . . . . ________

Field name     Field Qualifier
ADRECETT  +    ________________________________________________________  +
ADCPUT__  +    ________________________________________________________  +
ADDB2ETT  +    ________________________________________________________  +
________  +    ________________________________________________________  +
```

Figure 92. Specifying a Graph

Normally the graph will appear. If the graph cannot be displayed, messages are issued.

Occasionally messages are generated even though the graph is displayed. You can view these messages using the VIEWMSGS command after you have finished displaying the graph. The accounting and statistics graphs are plotted against time.

For more information about graphs showing data sharing information, refer to “Chapter 12. Reporting Data Sharing Information” on page 149.

DB2 PM uses the interactive chart utility (ICU) for graphics processing. For information on editing, printing, and saving graphs, refer to the corresponding Graphic Data Display Manager (GDDM(R)) documentation.

Common Accounting and Statistics Graph Elements

The DB2 PM accounting and statistics graphs have certain information in common. The following figure shows these common elements.

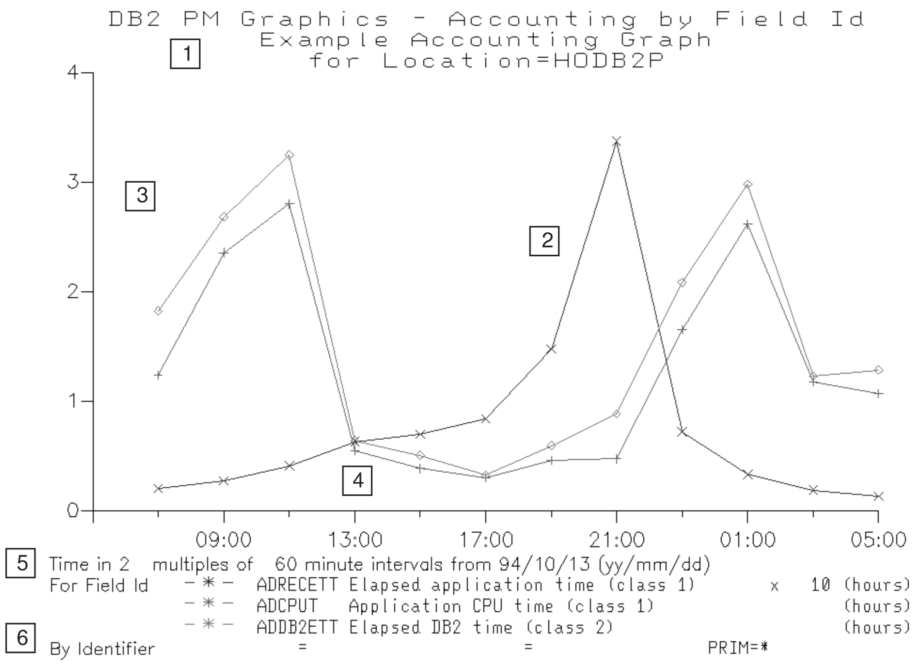

Figure 93. Sample Graph

1 *Title* —The title identifies the type of graph and shows the graph title you have specified. The last title line displays the name of the location from which the data originates.

2 *Graph* — The actual graph.

3 *Scales (Y-axis)* —The scale is adjusted according to the group of data having the highest value.

4 *Time (X-axis)* —The time axis can show either intervals or days, depending on what you select in the graph specification panel.

By interval: In background processing, the raw input data is summarized by time intervals. For example, data can be summarized into 60-minute intervals; thus, activity can be identified and plotted for each interval, one point every 60 minutes.

You can combine multiple input intervals to yield a single point. The number of intervals to be combined is specified in the *Number of units per division* option on the graph specification panel. This provides greater flexibility when plotting by time.

Graphing begins with the data for the earliest time interval after the requested From and To time. Consecutive time intervals (or summations of intervals) are plotted until the end of file is reached or until the stop date and time are reached.

Up to 30 X-axis points can be plotted. If the time selection and interval multiple combination yield more than 30 points, only the first 30 points are used. If a broader time range is required, the interval multiple can be increased.

The times of the X-axis reflect the beginning time for each interval or group of intervals. The date for the beginning point is a part of the X-axis title.

The interval version is graphed in line format. Alternative formats can be obtained via the interactive chart utility.

By day: The day version reports a single point for each graph in intervals of one or more days within the graphing period. Activity within the daily graphing interval is summed and plotted as a single point.

Note that although data is plotted in daily intervals, it does not have to cover the full 24 hours. You can request data from a specified time of the day by specifying a beginning and ending time. For example, you could plot the data from 9 a.m. to 5 p.m. for all the days in a week.

The day version is graphed in bar format. Alternative formats can be obtained via the interactive chart utility.

5 *X-axis Title* —The title below the X-axis describes the time selection on the X-axis. If multiple intervals are selected, specific additional information about the intervals is displayed. If day is selected, the From and To time per day is displayed.

6 The bottom part of the screen shows the field and DB2 PM identifiers used in the graph.

Field Identifiers —Identifies the field to be plotted. The IRF prompt facility (**F4**) on the graph specification panel provides lists of the appropriate field identifiers to be plotted.

When graphing by field identifier, a line is displayed after the X-axis title for each field identifier selected. Field Identifier is shown first, preceded by the '— * —' symbol indicating the corresponding group of data by color.

The unit of the Y-axis is displayed at the right corner of the screen. The units for times are:

(hours) Time in hours
(mins) Time in minutes
(secs) Time in seconds
(msecs) Time in milliseconds

If the data is scaled, the scaling factor is shown as:

(x 10) Value times 10
(x 100) Value times 100
(x 1000) Value times 1000
(x 10K) Value times 10 000

DB2 PM Identifiers —This line shows the selected DB2 PM identifier types and values. A maximum of three identifiers can be selected and displayed per line. If graphing by identifier, the '— * —' is displayed to indicate the corresponding data line by color. Equal signs (=) on the identifier line indicate that only one DB2 PM identifier was specified.

Accounting Graphs

You can produce two kinds of accounting graphs: accounting by field or accounting by DB2 PM identifier. Use the field identifier graphs to compare the values of different accounting fields (such as times, locking, and buffer pool information) and the DB2 PM identifier graphs to compare the value of one field for different DB2 PM identifiers (such as primary authorization IDs or plans).

Accounting by Field Identifier Graphs

You can use the accounting by field identifier graphs to analyze the performance of one group of transactions over a time interval to determine trends. You can plot up to four accounting fields for the group. For example, you may wish to graph the class 1 and class 2 times to determine the elapsed time pattern.

The following example plots a graph showing class 1 and class 2 times.

```
DGOGACFI        Graphics - Accounting by Field Identifiers

Update the following fields, then press Enter.

Graph title . . . . . . Example Accounting Graph______
SAVE data set . . . . . ACSAVDD___________________________________________
Location  . . . . . . . HODB2P______________
Group . . . . . . . . . ________

Field name     Field Qualifier
ADRECETT  +    ________________________________________________________  +
ADCPUT__  +    ________________________________________________________  +
ADDB2ETT  +    ________________________________________________________  +
________  +    ________________________________________________________  +

                        YY  MM  DD  HH  MM  SS  TH
From date/time  . . . . 94  10  13  07  00  00  00
To date/time  . . . . . 94  10  14  05  00  00  00

X-Axis divisions
2__  Number of units per division
2    Units (1=Days 2=Intervals)

DB2 PM Identifiers           Values
PRIM  +                     *
____  +                     ________________
____  +                     ________________
Command ===> ______________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Save      F7=Up
 F8=Down      F9=Swap     F11=Viewmsgs F12=Cancel
```

Figure 94. Specifying Accounting by Field Identifiers Graph

The following information was entered in the graph specification panel:

- The title *Example Accounting Graph*.
- The name of the accounting save data set.
- The location for which the graph is plotted. Only data for this location is shown in the graph.
- The group name is left blank because the example is for a non-data-sharing environment.
- The fields: elapsed application time (ADRECETT), application CPU time (ADCPUT), and elapsed DB2 time (ADDB2ETT).

  You can use **F4** (Prompt) to display a list of available fields. Some fields can be qualified by the package name, remote location, or buffer pool name.
- In this example, From and To dates and times were specified to get a period of the input data set only.
- The data is plotted using two intervals per unit of the x-axis.
- The default value for DB2 PM identifier PRIM (primary authorization ID) is *, so all data within the specified From/To time frame is used.

Figure 95 shows the resulting graph.

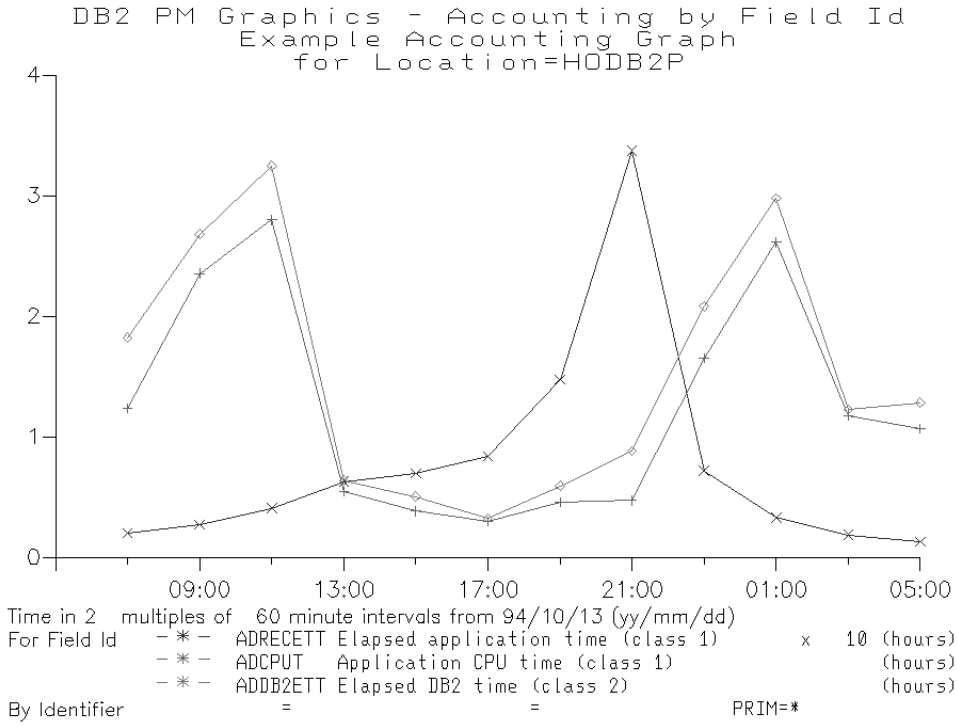

Figure 95. Accounting by Field Graph

Accounting by DB2 PM Identifier Graphs

You can use the accounting by DB2 PM identifier graphs to analyze the performance of up to four groups of transactions (combinations of, for example, primary authorization IDs and plan names) for one field over a period of time. This graph can be used in determining if one transaction is adversely affecting another.

In this example, the elapsed application times are compared for two different primary authorization IDs.

```
DGOGACDI          Graphics - Accounting by DB2 PM Identifier

Update the following fields, then press Enter.

Graph title . . . Example Accounting Graph______
SAVE data set . . ACSAVDD______________________________
Location  . . . . HODB2P___________________________
Group . . . . . . ________
Field name  . . . ADRECETT  +
Qualifier . . . . ________________________________________________________  +

                        YY  MM  DD  HH  MM  SS  TH
From date/time  . . . . __  __  __  __  __  __  __
To date/time  . . . . . __  __  __  __  __  __  __

X-Axis divisions
1    Number of units per division
2    Units (1=Days 2=Intervals)

DB2 PM Identifiers          Values
PRIM  +                    AGDFEL__________  HDAGR___________
                           ________________  ________________
____  +                    ________________  ________________
                           ________________  ________________
____  +                    ________________  ________________
                           ________________  ________________
Command ===> _________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Save      F7=Up
 F8=Down      F9=Swap     F11=Viewmsgs F12=Cancel
```

Figure 96. Specifying Accounting by DB2 PM Identifiers Graph

The following information was entered in the graph specification panel:

- The title *Example Accounting Graph*.
- The name of the accounting save data set.
- The location for which the graph is plotted. Only data for this location is shown in the graph.
- The group name is left blank because the example is for a non-data-sharing environment.
- The field: elapsed application time (ADRECETT).

  You can use **F4** (Prompt) to display a list of available fields. Some fields can be qualified by the package name, remote location, or buffer pool name.
- In this example, From or To dates and times were not specified, so all data in the input data set is used.
- The data is plotted using one interval per unit of the x-axis.
- DB2 PM identifier values: primary authorization IDs AGDFEL and HDAGR.

Figure 97 shows the resulting graph.

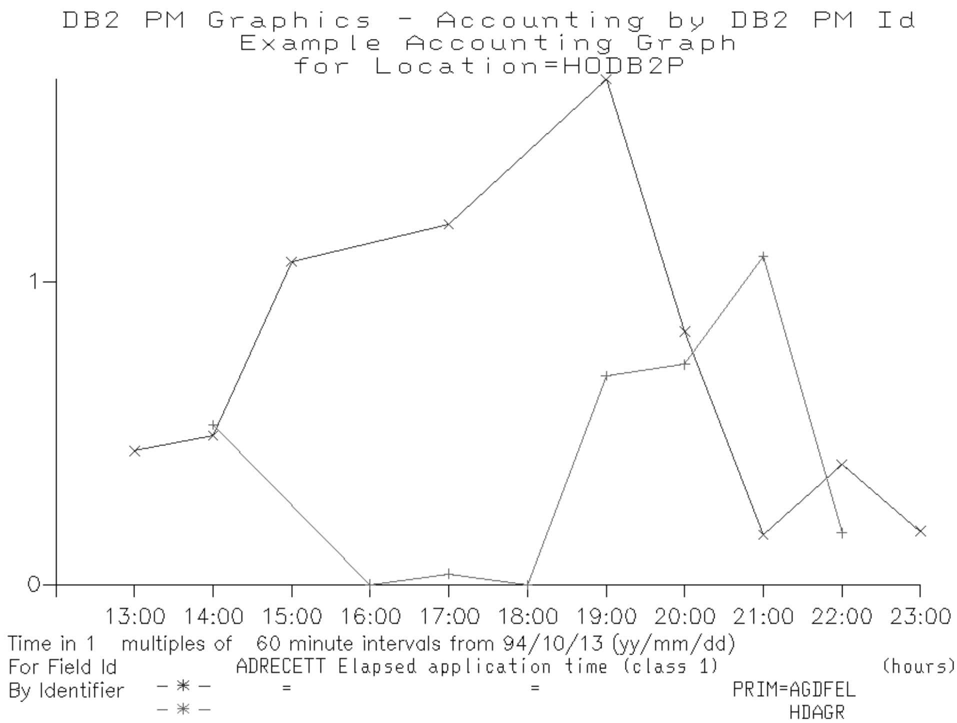

Figure 97. Accounting by Identifiers Graph

Statistics Graphs

Use the statistics graph to compare up to four statistics fields against time.

You can use the statistics graph to analyze the DB2 system performance over a period of time. This graph can be used to monitor system-wide DB2 resource usage trends.

For example, a graph like the following showing the system services address space TCB time, system services address space SRB time, and IRLM TCB time, might be useful.

```
DGOGSTAT                 Graphics - Statistics

Update the following fields, then press Enter.

Graph title . . . . . . Example Statistics Graph______
SAVE data set . . . . . STSAVDD______________________________
Location  . . . . . . . HODB2P___________________________
Group . . . . . . . . . ________

Field name     Field Qualifier
SSTCBT__  +    ________________________________________________________  +
SSSRBT__  +    ________________________________________________________  +
SDITCBT_  +    ________________________________________________________  +
________  +    ________________________________________________________  +

                        YY  MM  DD  HH  MM  SS  TH
From date/time  . . . . __  __  __  __  __  __  __
To date/time  . . . . . __  __  __  __  __  __  __

X-Axis divisions
1    Number of units per division
2    Units (1=Days 2=Intervals)

DB2 PM Identifiers          Values
SSID  +
      +

Command ===> ______________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Save      F7=Up
 F8=Down      F9=Swap     F11=Viewmsgs F12=Cancel
```

Figure 98. Specifying Statistics Graph

The following information was entered in the graph specification panel:

- The title *Example Statistics Graph*.
- The name of the statistics save data set.
- The location for which the graph is plotted. Only data for this location is shown in the graph.
- The group name is left blank because the example is for a non-data-sharing environment.
- The fields: system services address space TCB time (SSTCBT), system services address space SRB time (SSSRBT), and IRLM TCB time (SDITCBT).

  You can use **F4** (Prompt) to display a list of available fields. Some fields can be qualified by the package name, remote location, or buffer pool name.
- In this example, From or To dates and times were not specified, so all data in the input data set is used.

Figure 99 shows the resulting graph.

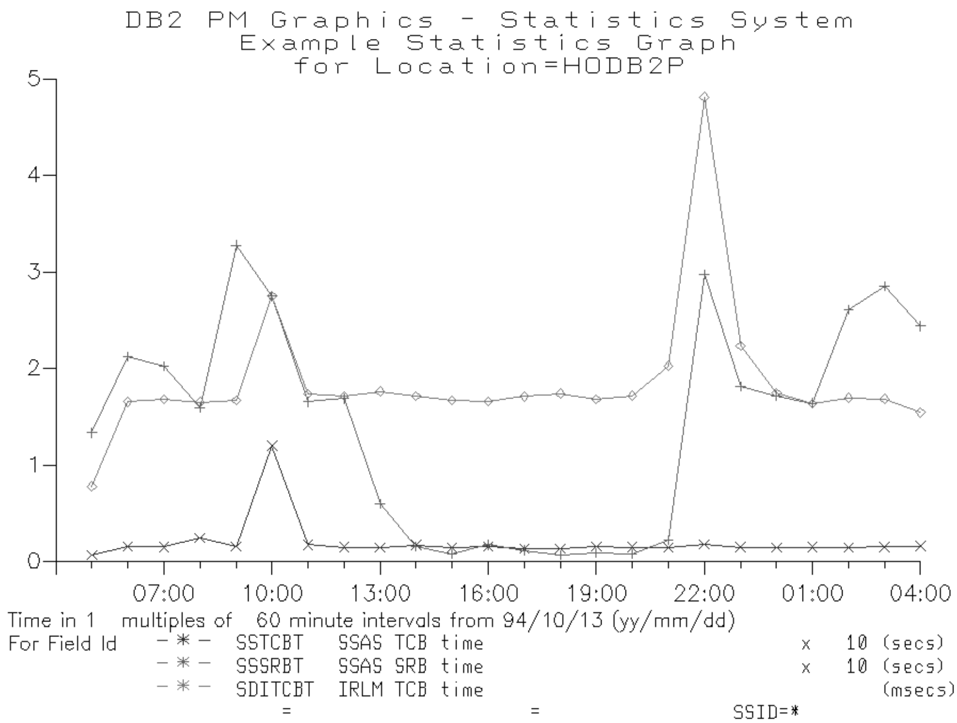

Figure 99. Statistics Graph

Frequency Distribution Graphs

Frequency distribution graphs plot the distribution of a selected field across user-specified ranges. The data used as input to the graph is produced using the DISTRIBUTE command.

Frequency distributions can help you locate unusual values by dividing events into ranges. This can be useful because in DB2 PM reports data is sometimes averaged over a large number of events, which makes the result difficult to analyze. An average value can make a small number of events that have exceptionally high or low values difficult to detect.

For example, you might notice large elapsed times in an accounting short report. Because reports show summarized data, you must either produce an accounting trace or use DISTRIBUTE and display the resulting frequency distribution graph. The graph can be the better alternative, because by specifying ranges for the elapsed times, you can quickly determine whether the majority of the application executions have a performance problem, and if so, how many of the application executions were in each elapsed time range.

DISTRIBUTE Command

Use the DISTRIBUTE command to specify the fields to be distributed and the frequency distribution limits to accumulate the data in appropriate ranges. After you have executed the command, the collected frequency distributions that have been written to the frequency distribution data set can be viewed using graphics.

You can specify the same field in multiple DISTRIBUTE commands, with different DB2 PM identifiers.

The DISTRIBUTE command is used for DB2 PM data that is associated with accounting, I/O activity, locking, SQL activity, and utility activity report sets.

As with all DB2 PM commands, DISTRIBUTE can be specified using the IRF or entering the command using an ISPF editor.

If you use the IRF, choose *Frequency Distributions* on the Interactive Report Selections panel.

Figure 100 shows how to make the specifications using the IRF DISTRIBUTE DB2 PM Identifier Selection panel.

```
DGOODNEW             DISTRIBUTE DB2 PM Identifier Selection

Enter a field name, qualifier (optional) and select up to three
DB2 PM identifiers from the list and press Enter to continue.

Field name . . ADRECETT  +
Qualifier  . . _______________________________________________________


__ __ __

              1   CONNECT  - Connection Identifier
              2   CONNTYPE - Connection Type
              3   CORRNAME - Correlation Name
              4   CORRNMBR - Correlation Number
              5   MAINPACK - Mainpack
              6   ORIGAUTH - Original Authorization Identifier
              7   PLANNAME - Plan Name
              8   PRIMAUTH - Primary Authorization Identifier
              9   REQLOC   - Requesting Location


Command ===> __________________________________________________________________
 F1=Help    F2=Split   F3=Exit    F4=Prompt  F7=Up      F8=Down    F9=Swap
F12=Cancel
```

Figure 100. DISTRIBUTE Specifications

On the DISTRIBUTE DB2 PM Identifier Selection panel specify the name of the field for which you want to produce frequency distributions. You can use **F4** (Prompt) to display a list of the available fields. Some fields can be qualified by the package name, remote location, buffer pool name, or resource limit facility type.

You can specify combinations of DB2 PM identifier values for which you want the frequency distribution data to be collected. The default is PRIMAUTH.

In the above example, ADRECETT (elapsed application time) was specified as the field to be distributed.

Press **Enter**. The Frequency Distribution Processing panel is displayed.

```
DGOODSTT         Frequency Distribution Processing

Complete the following, then press Enter.

Field Name . . . . : ADRECETT
Qualifier. . . . . :
Identifiers. . . . :

Units. . . . . . . . 1  1 Milliseconds
                        2 Seconds
                        3 Minutes
                        4 Hours
Limits . . . . . . . 1_________     5__________    10________
                     30________     50_________    100_______
                     300_______     500________    1000______










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

Figure 101. Specifying Ranges

Specify the units for the ranges and the actual ranges.

In this example, ranges of 0-1, 1-5, 5-10, 10-30, 30-50, 50-100, 100-300, 300-500, 500-1000 and more than 1000 milliseconds were specified for the elapsed application times field.

When you have specified all the ranges, press **Enter**. You can now specify the other information needed to run the job. For more information about using the IRF, refer to “Chapter 6. Running DB2 PM Jobs” on page 47.

If you do not want to use the IRF panels, you can use the ISPF/PDF editor to specify the DISTRIBUTE command using the following syntax:

```
⋮

DISTRIBUTE (FIELDID(ADRECETT)
            LIMITS(1 5 10 30 50 100 300 500 1000))
   ⋮
```

Refer to the *DB2 PM Report Reference* for a list of fields available for frequency distribution.

The data generated by DISTRIBUTE is, by default, written to the DD statement DISTDD.

Displaying Frequency Distributions

The following steps show how to produce a frequency distribution graph from the generated DISTRIBUTE data.

Select option 2 (*Display and print graphs*) from the DB2 PM main menu. The Graphics Selection Menu is displayed. To make the frequency distribution graph specifications, choose option 4. The Graphics - Frequency Distribution panel is displayed.

```
DGOGDIST             Graphics - Frequency Distribution

Update the following fields, then press Enter.

Graph title . . . Example Distribution Graph______
Data set  . . . . DISTSAVE________________
Location  . . . . HODB2D__________________
Group . . . . . . ________
Field name  . . . ADRECETT  +
Qualifier . . . . ________________________________________________________  +

                        YY  MM  DD  HH  MM  SS  TH
From date/time  . . . . __  __  __  __  __  __  __
To date/time  . . . . . __  __  __  __  __  __  __

DB2 PM Identifiers           Values
PRIM  +                     *
____  +                     ________________
____  +                     ________________



Command ===> _________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Save      F7=Up
 F8=Down      F9=Swap     F11=Viewmsgs F12=Cancel
```

Figure 102. Specifying Frequency Distribution Graph

The following information was entered in the graph specification panel:

- The title *Example Distribution Graph*.
- The name of the DISTRIBUTE data set. (This is the name of the data set you created as a result of executing the DISTRIBUTE command.)
- The location for which the graph is plotted. Only data for this location is shown in the graph.
- The group name is left blank because the example is for a non-data-sharing environment.
- The field: elapsed application time (ADRECETT).

  You can use **F4** (Prompt) to display a list of available fields.
- From or To dates and times were not specified, so all data in the input data set is used.
- No DB2 PM identifier values were specified for the field. The default value for DB2 PM identifier PRIM (primary authorization ID) is *, so all data in the input data set is used.

Figure 103 shows what the resulting graph looks like.

Graphs

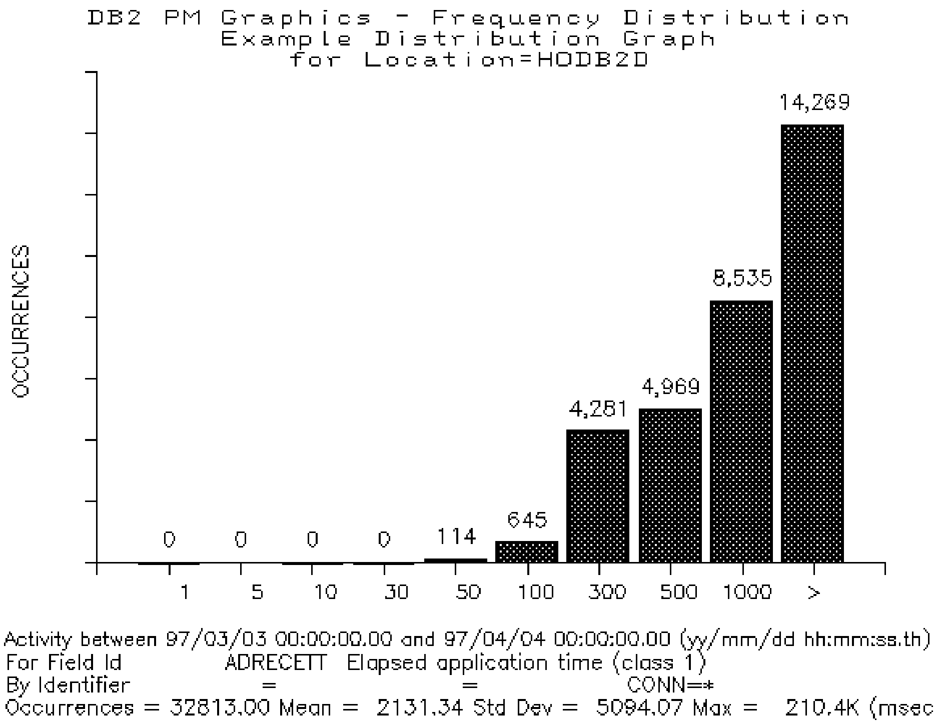

Figure 103. Frequency Distribution Graph

Notice that the following information is generated in addition to the frequency distribution graph:

- Number of occurrences
- Average value
- Standard deviation
- Maximum value.

Chapter 14. Common Errors Using DB2 PM

This chapter describes problems users often encounter when using DB2 PM. If you come across a problem, read this chapter before contacting IBM service.

When Generating DB2 PM Reports and Traces

This section describes problems encountered when generating DB2 PM reports and traces.

No Data to Report

Symptom: A "NO DATA TO REPORT" message is printed instead of a report.

Probable Cause: The cause is likely to be one of the following:

- The PRESORTED option was used on the GLOBAL command and the input data is not sorted.
- No data meets the INCLUDE/EXCLUDE criteria.
- No data resides between the specified FROM/TO times.
- The records required to create the requested report are not in the input data.
- The default REDUCE interval 0 was used with incompatible FROM specification on REPORT.

Explanation: There are several reasons why you may receive the above error message. DPMOUTDD cannot be specified as DUMMY. It can be omitted or made to point to a data set defined as described in the *DB2 PM Report Reference*.

If you use the PRESORTED option of the GLOBAL command, make sure your data has been previously sorted. If not, the job either halts with a return code of 8, or it completes with a return code of 0 and some or all of the input data may be lost.

INCLUDE/EXCLUDE and FROM/TO limit the data DB2 PM reports. If none of the data meets the INCLUDE/EXCLUDE or FROM/TO criteria, no report is produced. The timestamp of the DB2 records is the CPU store clock time when the record was produced. This may be the local time, GMT time, or any time when the CPU was set to. Therefore, report times may not always reflect the local time when the activity of interest occurred.

Also, REPORT FROM/TO times are applied to reduced records, which means that you must choose a suitable INTERVAL if you wish to filter by times. DB2 PM reports the next interval after the REPORT FROM time. If the default INTERVAL (0) is used (for example, if no REDUCE subcommand has been specified), the whole input data is reduced to one interval, starting with the timestamp of the first input record. If you specify a REPORT FROM time later than the first input record, DB2 PM reports no data, because there is no interval starting after the REPORT FROM time. In this case, specify the FROM/TO time with GLOBAL, which is processed before REDUCE.

The GLOBAL command parameter TIMEZONE can be used to adjust the DB2 record timestamps so that all reporting can be in local time. For more information, refer to "Specifying Time Zones" on page 126.

If the DB2 records needed to create a report are not in the input data, no report is produced. Refer to "Chapter 5. Collecting Performance Data" on page 27 or the general information chapter of the individual report sets in the *DB2 PM Report Reference* for a list of the IFCID records needed to produce a report and the DB2 traces that must be started to produce those records. If no report is created, check the IFCID frequency distribution log in the JOBSUMDD for the presence of the necessary IFCID records.

When interval 0 is used, all the records in the input data are stored into one interval with the timestamp of the first record. If the REPORT FROM time is later than the timestamp of the first record, DB2 PM does not find any data to report.

Unexpected Values on Reports and Traces

Symptom: Unexpected values are printed on reports and traces.

Probable Cause: The cause is likely to be one of the following:
- Data is being limited by the INCLUDE/EXCLUDE criteria.
- Data is being limited by the FROM/TO times.

Explanation: INCLUDE/EXCLUDE and FROM/TO times limit the data available for reporting to DB2 PM. Make sure these parameters are not excluding data you wish to report.

Unexpected Values in I/O Activity and Record Trace Output

Symptom: Sequential prefetch read I/O operations are not accounted for on the I/O activity or record trace output.

Probable Cause: Qualifying the plan or the authorization ID in the DB2 START TRACE command for performance trace class 4 restricts the data that DB2 produces to the user's task TCB, and omits all asynchronous tasks including sequential prefetch.

Explanation: DB2 does work on behalf of the thread under a separate agent or TCB. These asynchronous tasks are performed using a system level TCB and this work is not associated with, and is executed completely outside, the user's task TCB. Sequential prefetch is one of these asynchronous tasks.

When qualifying a DB2 START TRACE command for performance classes with a particular plan or authorization ID, DB2 does not produce sequential prefetch type activity in the trace data.

To ensure that all sequential prefetch related I/O is traced by DB2, and subsequently reported by DB2 PM, do not qualify the DB2 START TRACE command for read I/O operations by plan or by authorization ID. Note that you can qualify the DB2 PM I/O Activity report by plan name and authorization ID, so that you get only the report data you are interested in. This is possible because DB2 PM can relate the asynchronous I/O activity to the thread that initiated it.

Database and Page Set Names Not Translated

Symptom: The database and page set names are reported as decimal DBIDs or OBIDs.

Probable Cause: The data needed for translation is not available.

Explanation: The DB2 PM identifiers database name and page set name are frequently reported by DB2 PM. In most DB2 instrumentation records, the decimal DBID and OBID are recorded rather than the database or page set names. When possible, DB2 PM translates the DBID or OBID to the name of the database or page set. The information in IFCID 105 and IFCID 107 is used to perform the translation.

IFCID 105 is written for statistics class 1 and performance classes 1, 4, 6, 7, 8, 10, 13, and IFCID 107 for performance classes 1, 4, 6, 7, 8, 10, 13. IFCID 105 is written for a START TRACE command and at statistics intervals.

DB2 assigns the DBID or OBID when a database or page set is created. For all open databases and page sets, DB2 keeps a table containing DBIDs, OBIDs, and their corresponding database or page set names. The IFCID 105 record contains a copy of the table, recording the DBID and OBID translation to names, for all databases and page sets open at the time. When a data set is opened or closed, an IFCID 107 record is written, recording additions to, and deletions from, the table. At a given time, DB2 PM should have all the translation data for all open databases and page sets.

When DB2 PM encounters a DBID or OBID, it uses the translation data to translate the decimal ID to the corresponding character name, and reports the translated name. If the translation data is not available, the decimal ID is reported.

DB2 PM is unable to translate DBIDs and OBIDs when:
- The input data does not contain IFCID 105 or 107 records.
- There is no translation data for the particular DBID or OBID.
- The record is processed prior to the IFCID 105 or 107 record containing the necessary translation data.

Any report or trace that requires DBID/OBID translation needs a GLOBAL TO and FROM time range that crosses the time of writing statistics records (at the statistics interval) in order to be sure that IFCID 105 is present.

Make sure that the required IFCID records are collected in the input data. For example, when you start collecting GTF data, be careful not to filter out records using the JOBNAMP option.

The above considerations are especially important when producing locking traces based on the DB2 statistics trace. The input data should be limited to the appropriate statistics interval (for example 30 minutes), not to the exact time the deadlock occurred to make sure IFCID 105 is present.

When Using the IRF

This section describes problems encountered when using the IRF.

STEPLIB Missing from IRF-Generated JCL

Symptom: The STEPLIB DD statement is missing from the IRF-generated JCL.

Probable Cause: Wrong option selected while running DGOJVARS.

Explanation: DGOJVARS is an EXEC provided with DB2 PM that displays a panel for setting up certain DB2 PM defaults. One of the options on this panel specifies whether the DB2 PM load library is in the system LNKLST concatenation. If this option is selected, then no STEPLIB DD statement is in the IRF-generated JCL.

Command Not Found

Symptom: Message IKJ56500I `Command xxx not found` is issued during the execution of DB2 PM.

Probable Cause: The Program Control Facility (PCF) was not updated with the list of DB2 PM modules that are started as TSO command processors.

Explanation: The PCF makes a security check on all commands to see if they are included in the PCF command list. See your system administrator to ensure that all necessary updates have been made. After the list has been updated, the commands can be used.

Part 4. Monitoring and Tuning with DB2 PM Batch

Chapter 15. Which Reports to Run

Most sites always collect DB2 accounting and statistics trace data and regularly produce the following reports:
- Statistics
- Accounting
- System parameters.

The statistics reports show the system performance, the accounting reports show the thread activity, and the system parameters report shows the current configuration of the DB2 subsystem.

It is good practice to also generate deadlock and timeout reports regularly, because in this way detailed information is immediately available if you need to investigate a locking problem.

It is recommended that the statistics and accounting reports always be run using exception processing because exception reporting is the most effective means of highlighting actual and potential performance problems.

If a field is found in exception status and the exception reports do not provide enough information to determine the cause of it, the next step is to run statistics and accounting reports or traces. The vast majority of problems can be solved using the statistics and accounting reports in conjunction with one another. As an additional method of analysis, use TOP reporting to identify threads with the largest resource usage or the greatest wait time due to contention. TOP identifies these threads even if no exception conditions had been defined for them.

The more detailed report sets are needed only in those cases when the statistics and accounting report sets do not provide sufficient information about the cause of a problem.

To obtain the more detailed DB2 PM reports:
- Use the Online Monitor to collect the input data for your report. All you need to specify is the report you want to produce. Alternatively, you can start the DB2 performance trace with the appropriate classes and qualifiers to minimize the performance trace overhead.
- Recreate the problem, if possible.

Table 7 on page 178 lists examples of the situations where you might need to use the more detailed DB2 PM report sets to isolate the item causing performance problems. As elsewhere in this book, we have not included audit reports in the problem determination process, because they do not provide performance-related information.

Which Reports to Run

Table 7. When to Use the Detailed DB2 PM Reports

Problem	Found In	Go To
High number of deadlocks or timeouts	Statistics or accounting, deadlock or timeout traces	Locking reports
High class 2 TCB time	Accounting reports	EXPLAIN reports -> SQL activity reports
High I/O suspensions and high average number of synchronous reads	Statistics report, buffer pool information	Accounting reports -> EXPLAIN reports
High I/O suspensions but average number of synchronous reads OK	Accounting reports	System parameters and statistics (checkpoints and buffers) and RMF(TM) DASD I/O report
High I/O times, no other symptoms	Accounting	I/O activity reports
High lock/latch suspension time or high page latch wait	Locking reports	Lock suspensions -> accounting latch suspensions -> statistics
High synchronous read I/Os	SQL activity	EXPLAIN
Lock suspensions	SQL activity	Locking
High elapsed times for plan DSNBIND or DSNUTIL	Accounting	Utility activity reports

Where CP or Sysplex parallel processing is exploited, DB2 PM reports aggregate values across all parallel tasks created to execute SQL statements within the query. Because of the nature of parallel work, some counters (particularly CPU and suspension fields) might show unexpected large values, which do not necessarily indicate problems. In some cases these values can be larger than the entire wall-clock elapsed time.

Chapter 16. Monitoring Applications—Accounting

Tuning DB2

This chapter identifies and describes the specific DB2 data which is reported for the purpose of tuning DB2. For general tuning advice on DB2, refer to the DB2 Administration Guide ‘Performance, Monitoring, and Tuning’ chapters for the specific release of DB2.

This chapter shows how to interpret the key fields in the accounting reports.

The examples shown in this chapter are from a long accounting report instead of a trace, but you can find many of the same fields on the traces as well. Traces show application activity on a thread-by-thread basis, whereas on reports the data is summarized for DB2 PM identifiers. For example, if you have not changed the default ORDER parameters, the report entries show data summarized for every unique combination of primary authorization and plan, often including data from more than one thread.

Although the short report might be adequate for a majority of the DB2 sites, this chapter shows examples of the long report, because some DB2 sites might have a need to look at more data in their environment. The descriptions in this chapter cover most of the important fields on both the short and the long layouts.

The fields on the accounting reports are organized in blocks. You can decide which blocks and which fields within blocks you want to include on your report. For more information, refer to “Tailoring Report Layouts” on page 115.

For information on how to produce accounting reports and examples of different accounting reports, refer to “Accounting” on page 61. For information on how to add blocks and fields to an accounting report, refer to “Tailoring an Accounting Report” on page 116.

Identifying the Report Entry

Every report entry is identified by one to three fields showing the DB2 PM identifier values for which the data is summarized.

For example, if you have not changed the default order of plan name within primary authorization ID, this block of data consists of two fields, one showing the primary authorization ID, and the second the plan name in question.

If you have ordered the report by other DB2 PM identifiers, such as plan name or MAINPACK, this block shows the values of these identifiers. Refer to “Ordering the Data” on page 62 for information about summarizing data.

In the following example the default order of plan name within primary authorization ID is used.

```
PRIMAUTH: USRT001  PLANNAME: R31GFUO
```

Figure 104. Accounting Report—Identification Block

The following list shows some of the DB2 PM identifiers that are most commonly used to order reports.

ENDUSER
The end user’s user ID at the user’s workstation.

TRANSACT
The end user’s transaction name at the workstation.

WSNAME
The end user’s workstation name.

PRIMAUTH
The primary authorization ID from connection or signon. This is SYSOPR for MVS operator commands and DB2 system internal agents. The connection authorization exit and the signon authorization exit can change the primary authorization ID so that it is not the same as the original primary authorization ID. Distributed authorization ID translation can also change the primary authorization ID.

PLANNAME
The name of the plan produced during the bind process and used by DB2 to process SQL statements encountered during statement execution. Some examples are DSNUTIL for utility, DSNBIND for bind activity, DISTSERV for non-DB2 requesters’ DBATs, and the application plan name for CICS and IMS.

CORRNAME
The correlation name derived from the correlation ID.

CORRNMBR
The correlation number derived from the correlation ID.

For TSO, the correlation ID is the logon ID, while for Batch it is the job name. For CICS, connection type, thread type, and transaction ID constitute the correlation ID. For IMS, the PST number and the PSB name constitute the correlation ID.

MAINPACK
The DB2 PM identifier used to define a representative package within a plan. Refer to “Defining the MAINPACK Identifier” on page 133 for more information. This identifier can be used as an alternative to PLANNAME when ordering data. For example, because non-DB2 requester DBATs always have DISTSERV for the plan name, it is not useful to order by PLANNAME.

PACKAGE
The DB2 PM identifier consisting of location name, connection ID, and package ID. This identifier can be used as an ORDER keyword to aggregate data by packages regardless under which plan a particular package was executed. Refer to “DB2 PM Identifiers” on page 107 for an example.

INTERVAL
The DB2 PM identifier which can be used as an ORDER key word to order data according to reduction intervals. Refer to “Be Careful with INTERVAL” on page 111 for more details.

Elapsed Time Distribution Block

The elapsed time distribution block shows the distribution of the application time, in-DB2 time, and suspension time of the originating task. This block lets you recognize immediately where the application spent its time. For threads exploiting query parallelism, only the nonparallel part is taken into account.

```
ELAPSED TIME DISTRIBUTION
----------------------------------------------------------------
APPL   |========> 17%
DB2    |=======> 15%
SUSP   |=================================> 67%
```

Figure 105. Accounting Report—Elapsed Time Distribution Block

APPL The ratio of the application elapsed time, expressed as a percentage of the entire elapsed time.

DB2 The ratio of the elapsed DB2 time, expressed as a percentage of the entire elapsed time.

SUSP The ratio of the suspension time, expressed as a percentage of the entire elapsed time.

Class 2 Time Distribution Block

The class 2 time distribution block shows the distribution of the active-in-DB2 time, suspension time, and not-accounted time of the originating task. This block lets you recognize immediately where the application spent its time. For threads exploiting query parallelism, only the nonparallel part is taken into account.

```
CLASS 2 TIME DISTRIBUTION
----------------------------------------------------------------
CPU    |==> 4%
NOTACC |=======> 14%
SUSP   |========================================> 81%
```

Figure 106. Accounting Report—Class 2 Time Distribution Block

CPU The ratio of the DB2 CPU time, expressed as a percentage of the DB2 elapsed time.

NOTACC The ratio of the DB2 not-accounted time, expressed as a percentage of the DB2 elapsed time.

SUSP The ratio of the DB2 suspension time, expressed as a percentage of the DB2 elapsed time.

Highlights of Application Performance

The highlights block shows key information about the report entry, such as the number of accounting records that make up the entry, the types of threads included, and the average number of commits and rollbacks per occurrence.

```
HIGHLIGHTS
--------------------------
#OCCURRENCES    :        2
#ALLIEDS        :        1
#ALLIEDS DISTRIB:        0
#DBATS          :        1
#DBATS DISTRIB. :        0
#NO PROGRAM DATA:        0
#NORMAL TERMINAT:        2
#ABNORMAL TERMIN:        0
#CP/X PARALLEL. :        2
#IO PARALLELISM :        0
#INCREMENT. BIND:        0
#COMMITS        :        2
#ROLLBACKS      :        0
MAX SQL CASC LVL:        3
UPDATE/COMMIT   :     0.00
SYNCH I/O AVG.  :     0.00
```

Figure 107. Accounting Report—Highlights Block

#OCCURRENCES
The number of accounting records that make up this entry.

This counter is used (as a divisor) for calculating averages for class 2, class 3, and class 8 times and events.

#ALLIEDS
The number of threads that did not involve distributed activity.

#ALLIEDS DISTRIB
The number of threads that were initiated by a DB2 attach and requested data from one or more server locations.

It is important to know if the thread is involved in distributed activity, since this can affect the fields you should monitor. For example, if the thread is involved in distributed activity, the class 1 elapsed time is higher than it would be if there were no distributed activity, since this time includes VTAM time. Class 1 elapsed time is discussed later in “Class 1, Class 2, and Class 5 Times” on page 186.

#DBATS
The number of threads that were initiated, created, and performed work on behalf of a remote (requester) location.

#DBATS DISTRIB.
The number of threads that were initiated by a requester location to a server location that in turn requested data from another server location.

#NO PROGRAM DATA
The number of accounting records that do not contain package or DBRM data. If the value for this field is not zero, then it indicates that accounting trace classes 7 or 8 were not active for the entire period covered by the accounting report. This should be taken into consideration when interpreting package-related or DBRM-related data in the accounting report.

#NORMAL TERMINAT
The number of threads that terminated normally.

#ABNORMAL TERMIN
The number of threads that terminated abnormally.

#CP/X PARALLEL.
The number of accounting records where parallel query processing (which includes I/O parallelism) is used. The value of this field indicates the number of originating tasks.

#IO PARALLELISM
The number of accounting records where query I/O parallel processing is used.

#INCREMENT. BIND
The number of incremental binds.

If a plan is bound with VALIDATE(RUN), DB2 performs validity checks at bind time and rechecks any failures at run time. This can result in catalog contention and degraded application performance, depending on the number of statements flagged and how many times they are executed. Avoid VALIDATE(RUN) as much as possible. Ensure that all objects are created and all privileges are granted before bind, and select the VALIDATE(BIND) option.

In addition to plans bound with VALIDATE(RUN), this counter is incremented for plans using DB2 private protocol.

#COMMITS
The average number of commit phase 1, commit phase 2, read-only commit, and synchs.

#ROLLBACKS
The average number of rollbacks. This is the number of units that were backed out, including abends from attaches.

A nonzero value deserves investigation, as the rollback can be due to any of the following reasons:
- Application program abends
- Application rollback requests
- Application deadlocks on database records
- Application canceled by operator
- Thread abends due to resource shortage.

MAX SQL CASC LVL
The maximum level of indirect SQL cascading. This includes cascading due to triggers, UDFs, or stored procedures.

UPDATE/COMMIT
This ratio shows the inserts, updates, and deletes per commit and abend.

The ratio is useful in monitoring the frequency of commits. Too many updates per commit means a long running unit of recovery, which degrades IRLM performance and has repercussions for recovery in case of a failure. Too few updates per commit hampers performance, since log writes are involved. If necessary, look into the application to establish commit points appropriately.

The commit frequency can vary depending on when online transactions are executed and when batch jobs are executed.

SYNCH I/O AVG.
The average synchronous I/O time per event.

Response Time—Plan Level

Response time is usually the prime indicator of a performance problem and most often should be the starting point for analysis.

DB2 response times are classified as follows:

- Class 1 time shows the response time including time spent outside DB2.
- Class 2 elapsed time shows the time spent in DB2. It is divided into CPU time, other in DB2, and class 3 (wait) time.
- Class 3 elapsed time is divided into various waits, such as the duration of suspensions due to waits for locks and latches or waits for I/O.

Figure 108 shows the relationship between the different accounting times.

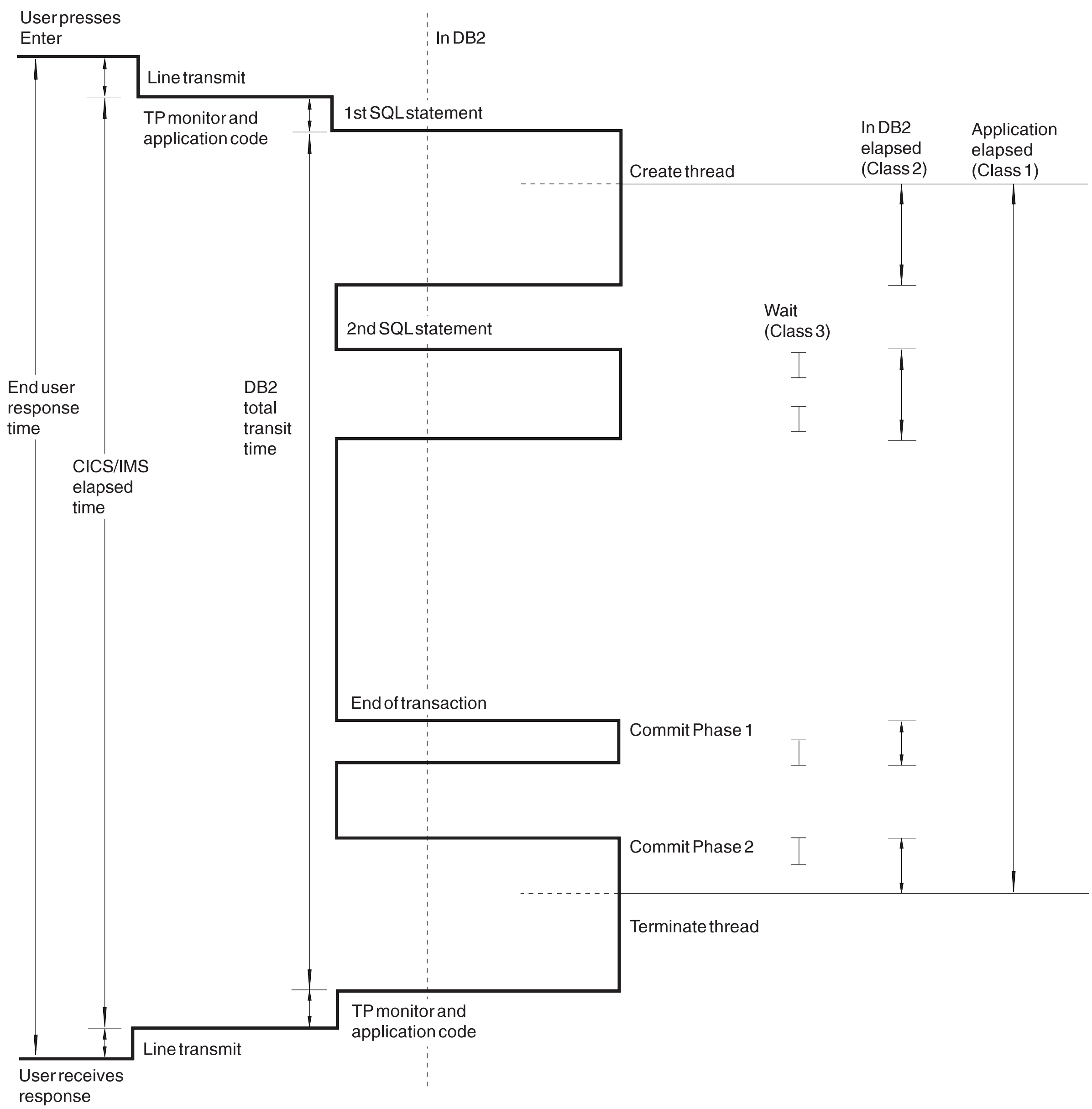

Figure 108. Accounting Times

If a query exploits CP and Sysplex parallel processing, several tasks (called parallel tasks) are scheduled to perform the parallel 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, which contains the details on nonparallel work within the thread as well as some parallel work-related data.

DB2 PM summarizes all accounting records generated for such a query and presents them as one logical accounting record. Table 8 on page 185 describes which values are a combination of originating records and parallel records and which are taken from the originating record only.

Especially interesting is the relationship between elapsed, CPU, and suspension times in the case of query CP and Sysplex parallel processing. The elapsed time is taken from the originating record while CPU and suspension times are calculated from all the 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 the response time distribution if the times for each participating task are known. Produce a long record trace for IFCID 3, especially if you suspect that the CPU times or suspension times for a thread where query CP and Sysplex parallel processing is used are large for other reasons than the times being added for several tasks. In a long record trace, all accounting records for originating and parallel tasks 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 several DB2 subsystems 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.

Table 8. Data Related to Query CP and Sysplex Parallel Processing

Accounting Data	Derivation
Identifiers (PRIMAUTH, PLANNAME, and so on)	Originating task
Class 1 elapsed time	Originating task
Class 1 CPU times	Separate counters for originating task and sum of all parallel tasks
Class 2 elapsed time	Originating task
Class 2 CPU times	Separate counters for originating task and sum of all parallel tasks
Class 7 elapsed time	Originating task
Class 7 CPU times	Separate counters for originating task and sum of all parallel tasks
Class 2 and class 7 DB2 entry/exit events	Originating task
Class 3 and class 8 times	Separate counters for originating task and sum of all parallel tasks
Class 3 and class 8 events	Sum of originating task and all parallel tasks
Class 5 times	Originating task
SQL counters	Originating task
RID List counters	Sum of originating task and all parallel tasks
Query Parallelism counters	Originating task

Table 8. Data Related to Query CP and Sysplex Parallel Processing (continued)

Accounting Data	Derivation
Locking (including data-sharing-specific) counters	Sum of originating task and all parallel tasks
RLF data	Originating task
Buffer Pools counters	Sum of originating task and all parallel tasks
Group Buffer Pools counters	Sum of originating task and all parallel tasks
DDF counters	Originating task
Data Capture counters	Originating task
SU counters	Separate counters for originating task and sum of all parallel tasks

Class 1, Class 2, and Class 5 Times

```
AVERAGE       APPL(CL.1)  DB2 (CL.2)  IFI (CL.5)
------------  ----------  ----------  ----------
ELAPSED TIME  1:01.61094  1:01.00275         N/P
 NONNESTED    1:01.61094  1:01.00275         N/A
 STORED PROC    0.000000    0.000000         N/A
 UDF            0.000000    0.000000         N/A
 TRIGGER        0.000000    0.000000         N/A

CPU TIME       56.879960   56.838591         N/P
 AGENT         13.210025   13.168926         N/P
  NONNESTED    13.210025   13.168926         N/A
  STORED PRC    0.000000    0.000000         N/A
  UDF           0.000000    0.000000         N/A
  TRIGGER       0.000000    0.000000         N/A
 PAR.TASKS     43.669665   43.669665         N/A

SUSPEND TIME         N/A    8.359907         N/A
  AGENT              N/A    1.310411         N/A
  PAR.TASKS          N/A    7.049495         N/A

NOT ACCOUNT.         N/A   46.523416         N/A
DB2 ENT/EXIT         N/A       19.00         N/A
EN/EX-STPROC         N/A        0.00         N/A
EN/EX-UDF            N/A        0.00         N/A
DCAPT.DESCR.         N/A         N/A         N/P
LOG EXTRACT.         N/A         N/A         N/P
```

Figure 109. Accounting Report—Application (Class 1), DB2 (Class 2), and IFI (Class 5) Times

Application Class 1

ELAPSED TIME
The class 1 elapsed time of the allied agent.

NONNESTED
The class 1 elapsed time for nonnested activity of the allied agent. This time does not include the time spent in stored procedures, user-defined functions, or triggers.

STORED PROC
The total elapsed time spent by the allied agent in stored

procedures. A stored procedure may initiate a trigger or invoke a user-defined function. The time spent there is not included in this counter.

UDF The total elapsed time spent by the allied agent in user-defined functions. A user-defined function may initiate a trigger or invoke a stored procedure. The time spent there is not included in this counter.

TRIGGER
The total elapsed time spent by the allied agent in triggers. A trigger may invoke a stored procedure or a user-defined function. The time spent there is not included in this counter.

CPU TIME
The class 1 CPU time in an application. It indicates:

- The class 1 CPU time of the allied agent, which might include the accumulated class 1 TCB time for processing stored procedures, user-defined functions, and triggers if present.
- The accumulated CPU time for processing parallel tasks. This is valid for query CP parallelism, Sysplex query parallelism, and parallel tasks generated by utilities.
- In Sysplex query parallelism, the individual CPU times are normalized by a conversion factor that is related to the originating task.
- In Sysplex query parallelism, only CPU times of parallel tasks running on the same member of the SYSPLEX group as the originating task, are included.

AGENT
The allied agent's class 1 CPU time in DB2. It comprises the class 1 CPU time of the allied agent, which might include the accumulated class 1 CPU time for processing stored procedures, user-defined functions, and triggers if present. The CPU time for processing parallel tasks is not charged to this counter.

NONNESTED
The class 1 CPU time of the allied agent's nonnested activity. It indicates the CPU time consumed by the allied agent for nonnested activity. This time does not include the time for processing stored procedures, user-defined functions, or triggers.

STORED PRC
The accumulated CPU time used to satisfy stored procedure requests processed in a stored procedure or WLM address space. This time is only calculated if accounting class 1 is active.

UDF The accumulated CPU time used to satisfy user-defined function requests processed in a stored procedure or WLM address space. This time is only calculated if accounting class 1 is active.

TRIGGER
The accumulated CPU time consumed while executing under control of triggers. For triggers, there is no distinction between class 1 and class 2 CPU time; all processing controlled by a trigger is within DB2.

PAR.TASKS
The CPU time accumulated in an application (class 1) for processing parallel tasks.

DB2 (Class 2)

ELAPSED TIME
The class 2 elapsed time for the allied agent accumulated in DB2 for the accounting record.

NONNESTED
The class 2 elapsed time for nonnested activity accumulated in DB2 for the allied agent. This time does not include the time spent in DB2 processing SQL statements issued by stored procedures, user-defined functions, or triggers.

STORED PROC
The total elapsed time that the allied agent spent executing SQL statements in stored procedures. A stored procedure may initiate a trigger or invoke a user-defined function. Any time spent there is not included in this counter.

UDF
The total elapsed time that the allied agent spent executing SQL statements in user-defined functions. A user-defined function may initiate a trigger or invoke a stored procedure. Any time spent there is not included in this counter.

TRIGGER
The total elapsed time spent by the allied agent in triggers. A trigger may invoke a stored procedure or a user-defined function. The time spent there is not included in this counter.

CPU TIME
The class 2 CPU time in DB2. It indicates:

- The class 2 CPU time of the allied agent, which might include the accumulated class 2 TCB time for processing stored procedures, user-defined functions, and triggers if present.
- The accumulated CPU time for processing parallel tasks. This is valid for query CP parallelism, Sysplex query parallelism, and parallel tasks generated by utilities.
- In Sysplex query parallelism, the individual CPU times are normalized by a conversion factor that is related to the originating task.
- In Sysplex query parallelism, only CPU times of parallel tasks running on the same member of the SYSPLEX group as the originating task, are included.

AGENT
The allied agent's class 2 CPU time in DB2. It comprises the class 2 CPU time of the allied agent, which might include the accumulated class 2 CPU time for processing stored procedures, user-defined functions, and triggers if present. The CPU time for processing parallel tasks is not charged to this counter.

NONNESTED
The class 2 CPU time of the allied agent's nonnested activity. It indicates the CPU time the allied agent spent in DB2 for nonnested activity. This time does not include the time for processing SQL statements issued by stored procedures, user-defined functions, or triggers.

STORED PRC
The accumulated CPU time consumed in DB2 processing SQL

statements issued by stored procedures. This time is only calculated if accounting class 2 is active.

UDF The accumulated CPU time consumed in DB2 processing SQL statements issued by user-defined functions. This time is only calculated if accounting class 2 is active.

TRIGGER
The accumulated CPU time consumed while executing under control of triggers. For triggers, there is no distinction between class 1 and class 2 CPU time; all processing controlled by a trigger is within DB2.

PAR.TASKS
The CPU time accumulated in a DB2 (class 2) for processing parallel tasks.

SUSPEND TIME
This is the same as TOTAL CL3 in Figure 110 on page 190. These fields show the time spent for suspension. The suspension time is the sum of suspensions for the agent and parallel tasks.

AGENT
The waiting time of the allied agent for all types of class 3 suspensions. This counter comprises class 3 suspensions within nested activity. Suspension time of parallel tasks in query or utility parallelism is not included.

PAR.TASKS
The accumulated suspension time spent for parallel tasks.

NOT ACCOUNT.
The time not accounted in DB2. Use this time to determine whether there is a large percentage of time that has not been captured within the DB2 accounting record.

Wait Times (Class 3)

Note: For threads exploiting query CP and Sysplex parallel processing, wait times are summed for the originating task and all parallel tasks. See also the discussion at the beginning of “Response Time—Plan Level” on page 184.

```
CLASS 3 SUSPENSIONS   AVERAGE TIME  AV.EVENT
--------------------  ------------  --------
LOCK/LATCH(DB2+IRLM)      1.500181      1.09
SYNCHRON. I/O             0.002096      0.13
 DATABASE I/O             0.000810      0.00
 LOG WRITE I/O            0.001286      0.00
OTHER READ I/O            0.000000      0.00
OTHER WRTE I/O            0.000000      0.00
SER.TASK SWTCH            0.860814      1.04
 UPDATE COMMIT            0.000000      0.00
 OPEN/CLOSE               0.459010      3.20
 SYSLGRNG REC             0.193708      0.91
 EXT/DEL/DEF              0.160772      0.31
 OTHER SERVICE            0.047324      0.26
ARC.LOG(QUIES)            0.000000      0.00
ARC.LOG READ              0.000000      0.00
STOR.PRC SCHED            0.129187      0.04
UDF SCHEDULE              0.000000      0.00
DRAIN LOCK                0.000000      0.00
CLAIM RELEASE             0.000000      0.00
PAGE LATCH                0.000000      0.00
NOTIFY MSGS               0.000000      0.00
GLOBAL CONT.              0.340642      7.37
FORCE-AT-COMMIT           0.000000      0.00
TOTAL CLASS 3             2.832920      9.67
```

Figure 110. Accounting Report—Suspension/System Times and Events (Class 3)

Class 3 provides the breakdown of the various waits.

LOCK/LATCH(DB2+IRLM)
This field shows the average duration of suspensions resulting from a lock or latch not being available immediately on request.

If the suspension time is high, examine further the locking block in accounting. If lock suspension is high, check the locking block. If latch suspension is high, check latch suspension statistics.

SYNCHRON. I/O
This field shows the average accumulated I/O elapsed wait time for I/O done under this thread. This field is for synchronous I/O only. It includes synchronous read and write I/O. If the time per I/O is high, one typical problem is an interference by prefetch or deferred write I/O.

DATABASE I/O
Average time: The accumulated I/O elapsed wait time for database I/O done under this thread. This field is for synchronous I/O only. It includes synchronous read and write I/O. This value indicates the elapsed time that the allied agent waited for its I/O in DB2. This value is an average.

Average event: The number of wait trace events processed for waits for database I/O under this thread. This value is an average.

LOG WRITE I/O
Average time: The accumulated wait time for log write I/O. This value is an average.

Average event: The number of wait trace events processed for waits for log write I/O. This value is an average.

OTHER READ I/O
This field includes waits caused by sequential prefetch, list prefetch, dynamic prefetch, and synchronous read I/O performed by other threads.

If the value in this field is high, the problem could be an I/O bound query using prefetch or an I/O contention. If it is an I/O bound query, DEGREE ANY could drastically improve elapsed time of such a query. The application is accessing data from a busy data set/volume/control unit and is continually being suspended. The DBA and the OS/390 systems programmer should be consulted.

OTHER WRTE I/O
Included in this field are waits caused by asynchronous write I/O and synchronous write I/O performed by other threads.

If the value in this field is high, the problem could be I/O contention. The application is accessing data from a busy data set/volume/control unit and is continually being suspended. The DBA and the OS/390 systems programmer should be consulted to resolve possible data set placement problems.

Too small a buffer pool for sort and other updated data could also cause this problem.

SER.TASK SWTCH
The accumulated waiting time due to a synchronous execution unit switching to DB2 services from the thread being reported. This value is an average.

UPDATE COMMIT
Average time: The accumulated wait time due to a synchronous execution unit switch for DB2 COMMIT, ABORT, or deallocation processing. This value is an average.

Average event: The number of wait trace events processed for waits for synchronous execution unit switching for COMMIT or ABORT. This value is an average.

OPEN/CLOSE
Average time: The accumulated wait time for a synchronous execution unit switch to the DB2 OPEN/CLOSE data set service or the HSM recall service. This value is an average.

Average event: The number of wait trace events processed for waits for synchronous execution unit switching to the OPEN/CLOSE service. This value is an average.

SYSLGRNG REC
Average time: The accumulated wait time for a synchronous execution unit switch to the DB2 SYSLGRNG recording service. Sometimes this service is also used for level ID checking for down-level detection.

Average event: The number of wait trace events processed for waits for synchronous execution unit switching to the SYSLGRNG recording service. This value is an average.

EXT/DEL/DEF
Average time: The accumulated wait time for a synchronous execution unit switch to the DB2 data space manager services, which include DEFINE DATA SET, EXTEND DATA SET, DELETE DATA SET, RESET DATA SET, and VSAM catalog access. This value is an average.

Average event: The number of wait trace events processed for waits for synchronous execution unit switching to the data space manager service tasks. This value is an average.

OTHER SERVICE
Average time: The accumulated wait time for a synchronous execution unit switch to other DB2 service tasks. This value is an average.

Average event: The number of wait trace events processed for waits for synchronous execution unit switching to other service tasks. This value is an average.

ARC.LOG (QUIES)
Included here are waits due to processing of Archive command (not the time for the Archive command to complete).

Avoid issuing the -ARCHIVE LOG QUIESCE during peak periods.

ARC.LOG READ
Accumulated wait time for archive reads (from tape).

STOR.PRC SCHED
The time spent waiting for the stored procedure to be scheduled, and the number of times an SQL CALL waited for a procedure to be scheduled.

UDF SCHEDULE
The UDF schedule.

DRAIN LOCK
The time spent waiting due to drain lock suspensions. If the value in this field is large, it might indicate that the execution of a utility impacted the SQL.

CLAIM RELEASE
The time spent waiting for claims to be released.

PAGE LATCH
The time spent waiting for page latch contentions.

NOTIFY MSGS
The time spent waiting to send notify messages to other members, in the data sharing group, and the number of times the agent was suspended due to sending messages to other members.

GLOBAL CONT.
The time spent waiting due to global lock contention in a data sharing environment, and the number of times the agent was suspended due to group-level contentions.

FORCE-AT-COMMIT
Average time: The accumulated time spent waiting for force-at-commit. This value is an average.

Average event: The number of wait trace events for force-at-commit. This value is an average.

TOTAL CLASS 3
Total time spent waiting because of various suspensions covered by accounting trace class 3.

CLASS 1, CLASS 2, AND CLASS 3 TIME CONSIDERATIONS

Compare the class 1 elapsed time with the CICS/IMS transaction times from CICS/IMS monitor:

Where query parallelism is exploited, DB2 schedules the *originating* and several *parallel* tasks. The sum of the CPU and suspension times used by these tasks may often exceed the wall-clock elapsed time for the query. Where Sysplex query parallelism is exploited, the CPU times represent normalized times. The following can happen:

- Total class 2 CPU might exceed parent class 2 elapsed time.
- Total class 3 suspensions might exceed parent class 2 elapsed time.

This should always be taken into consideration when analyzing times. It does not necessarily indicate problems. If additional information on parallel tasks is required, run the Record Trace report, which provides details of individual tasks.

- If the class 1 elapsed time is significantly less than the transaction time, analyze the CICS or IMS monitoring information first to find out the reason for the above. If the CICS/IMS information does not provide an answer to the problem, DB2 performance trace can be started and SQL activity reports generated. The elapsed time does not include the thread creation and thread termination times nor the time before the first SQL call. Check the SQL activity report to see if the thread creation and termination times are reasonable, and determine whether the problem is in DB2 or the transaction monitor. A large difference can also occur if there is a lot of processing going on before the first call to DB2 or after terminating the DB2 thread.
- If the class 1 elapsed time is significantly more than the CICS or IMS monitor elapsed times, it shows that the transaction is either a CICS protected thread or an IMS WFI thread. In both these cases, the thread might be waiting around for a transaction to use it, and will not be doing any work. In CICS and IMS monitors, waiting for a transaction time is not reported, but the DB2 accounting trace data does report it. Another point to note is that if the thread is involved in distributed activity, this time includes VTAM time. Note also that inactive DBATs do not continue to accrue class 1 elapsed time while they are inactive.

CLASS 1, CLASS 2, AND CLASS 3 TIME CONSIDERATIONS continued

Compare the class 2 elapsed time with the class 1 elapsed time:

- The difference between class 1 elapsed time and class 2 elapsed time is the time that is spent outside of DB2. This is sometimes called NOT in DB2 time. If the difference is significant, the problem could be in the application program, CICS, or IMS. However, it shows that the problem is not a DB2 problem.
- Note that for CICS protected threads and IMS WFI transactions, the thread is not terminated when a transaction completes. The class 1 elapsed time is therefore meaningless, as it might reflect the delay until next signon, resignon, or thread termination.

Compare the class 2 CPU time with the class 1 CPU time:

- The difference between the class 1 CPU and the class 2 CPU can be classified as the CPU time used for the application processing occurring outside of DB2.
- It provides a ratio of application processing time to DB2 processing time for an application. Note, however, that this consideration does not apply to CICS applications.
- The ratio can be used as a guideline for determining if the application processing has altered.

Compare the class 2 elapsed time with the class 2 CPU time:

- If the difference between the elapsed and CPU times for class 2 is high, there is a lot of wait time due to I/O, lock/latch suspensions, or other DB2 suspensions. Note that for threads exploiting query CP and Sysplex parallel processing, special considerations apply. See also the discussion at the beginning of “Response Time—Plan Level” on page 184.

  Class 3 times need to be examined. If they account for the difference, no need to go further. Otherwise look at the not accounted time in DB2 shown in the field **NOT ACCOUNT.**. CPU Wait time, Paging, and so on can contribute to this.

CLASS 1, CLASS 2, AND CLASS 3 TIME CONSIDERATIONS continued

Examine Lock and Latch suspensions (Class 3):

- If there are a large number of lock suspensions, analyze the Locking Report Set to determine the reason for the suspensions and on which DB2 object the suspensions occurred. It is only when this has been determined that solutions can be considered based on the type of DB2 object and the SQL activity of the applications being suspended.

  Page latch suspensions are also associated with specific database objects and can be analyzed via the locking report set.

- Latch suspensions are generally quite short. The latch suspensions reported in this field include IRLM latch suspensions as well as DB2 latch suspensions for latches acquired internally by various DB2 resource managers. Note that the DB2 code latch suspensions are not included in the latch suspension count of the locking activity block and therefore it is not expected that the counts are the same.

  The Lock/Latch suspension field is an ambiguous reference because of the addition of page latch wait time.

  The IRLM latches are held during the deadlock detection cycle, which is set by the user. The fewer locks an application takes, the fewer latches there are.

- If there have been timeouts or deadlocks, there is a contention problem where locks are not being released quickly enough before the timeout occurs. The lockout trace should be analyzed to determine which applications were holding the locks required.

  The timeout trace and deadlock trace as well as reports can be readily obtained since this requires Statistics class 3 trace, which is recommended to be always active.

Once the problems are located, a number of factors can be considered to reduce such problems. These include changing the lock type on a table, increasing the amount of free space per page, adding more commits to an application, or changing the application, as well as system-wide tuning of locking parameters.

CLASS 1, CLASS 2, AND CLASS 3 TIME CONSIDERATIONS continued

Examine synchronous I/O suspensions (Class 3):

If the I/O suspension time is high and the number of synchronous read I/Os is higher than expected,

- There may have been a change in the access path for the transaction. You will probably also notice a high number of GETPAGEs should there be a change in the access path.
- There could be a system-wide buffer pool or an EDM pool problem, or a combination of these problems. It is possible for some prefetched pages to be written over in the buffer pool before the application had a chance to process them, if the buffer pool is not sufficiently large. The application is then forced to read the pages synchronously.
- There may be a need to reorganize the table space or index.

If the I/O suspension time is high and the number of synchronous read I/Os is not higher than expected:

- Check the number of synchronous write I/Os. Ideally, there should be little synchronous write I/Os. System tuning is required if there are many such write I/Os.

If the I/O suspension time is high and the number of synchronous read I/Os or synchronous write I/Os is not higher than expected:

- The problem could be I/O contention. The application is accessing data from a busy data set/volume/control unit and is continually being suspended. The DBA and the OS/390 systems programmer should be consulted, to resolve possible data set placement problems.
- The problem could be CPU contention. After a read I/O or write I/O completes, the application is not being dispatched until much later. The DBA and the OS/390 systems programmer should be consulted for a possible dispatching priority problem.
- The problem could be application wait due to log buffer force-write. Log buffer force-writes occur at commit time. Examine if the frequency of commits is high, and consider reducing the frequency if desired.

To get the true I/O suspension figures, it is necessary to go to GTF/RMF reports, as there may be some time between when the I/O is complete and when the application is finally dispatched.

Response Time by Packages (Class 7 and Class 8)

These two blocks of information contain response and wait time information for packages. The data is obtained by turning on accounting trace class 7 for the elapsed and CPU times, and class 8 for wait times. Refer to “Response Time—Plan Level” on page 184 for a discussion on how to interpret the time values for threads exploiting query CP and Sysplex parallel processing. These considerations apply also to class 7 times (for elapsed times and CPU times) and to class 8 times (for suspension times).

Package Times (Class 7 and Class 8)

This block of data shows the response times per package.

```
DGO@YX00                   TIMES
------------------  ------------
ELAP-CL7 TIME-AVG       7.429787
CPU TIME                0.108954
  AGENT                 0.108954
  PAR.TASKS             0.000000
SUSPENSION-CL8          2.864060
  AGENT                 2.864060
  PAR.TASKS             0.000000
NOT ACCOUNTED           4.456774
AVG.DB2 ENTRY/EXIT         10.00
DB2 ENTRY/EXIT                10

CPU SERVICE UNITS          59.00
  AGENT                    59.00
  PAR.TASKS                 0.00
```

Figure 111. Accounting Report—Package Activity

ELAP-CL7 TIME-AVG
These fields show the elapsed time and the CPU time for the processing performed in DB2 only. Elapsed time includes Wait times.

CPU TIME
This is the sum of the TCB times for the originating and parallel (PAR.TASKS) threads. For class 7, there is no separate field for TCB time from stored procedures. The class 7 time from a stored procedure is accumulated in the existing class 7 TCB time field.

AGENT
The class 7 CPU time of the allied agent for executing the package or DBRM in DB2. In query CP and and Sysplex query parallelism, this does not include the class 7 time for parallel tasks.

PAR.TASKS
This is the accumulated TCB time spent processing parallel tasks.

SUSPENSION-CL8
This is the same as TOTAL CL8 SUSPENS in Figure 112 on page 198. It is the sum of suspension times for the originating (TCB) and parallel (PAR.TASKS) thread.

AGENT
The allied agent’s class 8 suspension time for executing the package or DBRM in DB2. Suspension time of parallel tasks in query or utility parallelism is not included.

PAR.TASKS
The accumulated suspension time for all parallel tasks.

NOT ACCOUNTED
This is derived as (WAITING - SUSPENSION CL8). CPU Wait time, Paging, Open/Close activity can contribute to this.

AVG.DB2 ENTRY/EXIT and DB2 ENTRY/EXIT
The number of DB2 entry or exit events processed while executing this package or DBRM.

CPU SERVICE UNITS
The CPU service unit time for a package or DBRM.

AGENT
The allied agent’s class 7 CPU service unit time for executing the package or DBRM in DB2.

PAR.TASKS
The CPU service unit time accumulated for a package or DBRM for processing parallel tasks.

Package Suspensions (Class 8)

```
DGO@YX00            AVERAGE TIME  AVG.EV    TIME/EVENT
------------------  ------------  ------  ------------
LOCK/LATCH              0.000000    0.00           N/C
SYNCHRONOUS I/O         0.000000    0.00           N/C
OTHER READ I/O          0.000000    0.00           N/C
OTHER WRITE I/O         0.000000    0.00           N/C
SERV.TASK SWITCH        0.000000    0.00           N/C
ARCH.LOG(QUIESCE)       0.000000    0.00           N/C
ARCHIVE LOG READ        0.000000    0.00           N/C
DRAIN LOCK              0.000000    0.00           N/C
CLAIM RELEASE           0.000000    0.00           N/C
PAGE LATCH              0.000000    0.00           N/C
STORED PROCEDURES       0.000000    0.00           N/C
NOTIFY MESSAGES         0.000000    0.00           N/C
GLOBAL CONTENTION       0.000000    0.00           N/C
```

Figure 112. Accounting Report—Package Activity

This block of data shows the wait times per package.

LOCK/LATCH
This field shows the duration of suspensions resulting from a lock or latch not being available immediately on request.

If the suspension time is high, further examine the locking block in accounting. If lock suspension is high, check locking block. If latch suspension is high, check latch suspension statistics.

SYNCHRONOUS I/O
This field shows the accumulated I/O elapsed wait time for I/O done under this thread. This field is for synchronous I/O only. It includes synchronous read and write I/O and log write commit I/O. If the time per I/O is high, one typical problem is an interference by prefetch or deferred write I/O. I/O scheduling enhancement in DB2 Version 3 should drastically reduce this problem (by assigning higher I/O priority to synchronous I/O over asynchronous I/O).

OTHER READ I/O

This field includes waits caused by sequential prefetch, list prefetch, dynamic prefetch, and synchronous read I/O performed by other threads.

If the value in this field is high, the problem could be an I/O bound query using prefetch or an I/O contention. If it is an I/O bound query, DEGREE ANY could drastically improve elapsed time of such a query. The application is accessing data from a busy data set/volume/control unit and is continually being suspended. The DBA and the OS/390 systems programmer should be consulted.

OTHER WRITE I/O

This field includes waits caused by asynchronous write I/O and synchronous write I/O performed by other threads.

If the value in this field is high, the problem could be I/O contention. The application is accessing data from a busy data set/volume/control unit and is continually being suspended. The DBA and the OS/390 systems programmer should be consulted to resolve possible data set placement problems.

SERV.TASK SWITCH

Included here are waits due to OPEN/CLOSE data set, SYSLGRNX update (prior to DB2 Version 4: SYSLGRNG update), COMMIT PHASE II for UPDATE threads, HSM recall for data set, and data space manager services, DEFINE, EXTEND, and DELETE data set.

There is no overlap between the elapsed time reported in this field and the other class 3 elapsed times. For DB2 Version 4 and later, if service task suspensions overlap other types of suspensions, the other types of suspensions are ignored. Previously the service task time was ignored.

Preformatting the data sets is probably the most important in terms of service task suspension.

ARCH.LOG(QUIESCE)

Included here are waits due to processing of Archive command (not the time for the Archive command to complete).

Avoid issuing the -ARCHIVE LOG QUIESCE during peak periods.

ARCHIVE LOG READ

Accumulated wait time for archive reads (from tape).

DRAIN LOCK

The time spent waiting due to drain lock suspensions. If the value in this field is large, it might indicate that execution of a utility impacted the SQL.

CLAIM RELEASE

The time spent waiting for claims to be released.

PAGE LATCH

The time spent waiting for page latch contentions.

STORED PROCEDURES

The time spent waiting for the stored procedure to be scheduled, and the number of times an SQL CALL waited for a procedure to be scheduled.

NOTIFY MESSAGES

The time spent waiting to send notify messages to other members, in the data sharing group, and the number of times the agent was suspended due to sending messages to other members.

GLOBAL CONTENTION
The time spent waiting due to global lock contention in a data sharing environment, and the number of times the agent was suspended due to group-level contentions.

Termination Data for Normal Termination

This block indicates the reasons why the accounting records were written when the thread terminated normally.

```
NORMAL TERM.      AVERAGE     TOTAL
---------------  --------  --------
NEW USER             0.00         0
DEALLOCATION         1.00         2
APPL.PROGR. END      0.00         0
RESIGNON             0.00         0
DBAT INACTIVE        0.00         0
RRS COMMIT           0.00         0
```

Figure 113. Accounting Report—Normal Termination Block

In the case of CICS and IMS, this block also indicates whether the thread was in fact not terminated but reused. Further, in the case of CICS, it indicates whether there was a resignon (or partial signon). Resignon means that the accounting record is produced at thread reuse, even if the user authorization ID does not change. This happens when for the TYPE=INIT macro TOKENI=YES is specified in the RCT, or for the TYPE=ENTRY macro TOKENE=YES is specified in the RCT.

NEW USER
Thread is reused at SIGNON (CICS) and an accounting record is cut because the user authorization ID has changed.

Thread is reused at SIGNON (IMS) and an accounting record is cut with a reason of new user even when the authorization ID does not change on thread reuse.

DEALLOCATION
Thread is deallocated (no thread reuse). This applies also to CICS protected threads that are terminated after the purge cycle if no transaction eligible to reuse the thread has been received.

APPL.PROGR. END
Thread is deallocated (no thread reuse). This occurs when the application program terminates without using DB2 protocols to end its connection to DB2.

RESIGNON
Resignon (or partial signon) refers to the case when a thread is reused at SIGNON (CICS) and an accounting record is cut even though the user authorization ID has not changed. It also indicates that the attachment facility has passed the CICS LU 6.2 token to DB2 for inclusion in the DB2 accounting trace records making it easier to correlate DB2 and CICS accounting and trace records.

DBAT INACTIVE
A database access thread goes inactive. When an active database access thread commits or rolls back, and the ‘DDF THREADS =======> INACTIVE’ option is selected on the DB2 Distributed Data Facility install panel DSNTIPR, the thread becomes inactive if it holds no database locks

and does not have any cursors open with the HOLD option. When the database access thread becomes inactive, the accounting record is written.

RRS COMMIT
The number of times an RRS application with accounting interval specified as COMMIT ended normally.

In a TSO environment, the value is always DEALLOCATION or APPL.PROGR. END for normal termination since a thread cannot be reused.

If the environment is IMS and you do not see NEW USER, or the environment is CICS and you do not see either NEW USER or RESIGNON, then you may want to consider defining IMS or CICS parameters for the DB2 attachment to promote thread reuse.

Locking Activity

This block shows information about the locking activity for this report entry.

```
LOCKING                 AVERAGE    TOTAL
---------------------- -------- --------
TIMEOUTS                   0.00        0
DEADLOCKS                  0.00        0
ESCAL.(SHARED)             0.00        0
ESCAL.(EXCLUS)             0.00        0
MAX PG/ROW LOCKS HELD      6.00        6
LOCK REQUEST              31.00       62
UNLOCK REQUEST             5.50       11
QUERY REQUEST              0.00        0
CHANGE REQUEST             1.50        3
OTHER REQUEST              0.00        0
LOCK SUSPENSIONS           0.00        0
IRLM LATCH SUSPENSIONS     0.00        0
OTHER SUSPENSIONS          0.00        0
TOTAL SUSPENSIONS          0.00        0
```

Figure 114. Accounting Report—Locking Activity

TIMEOUTS
The number of times lock suspension ultimately resulted in a timeout. This happens when a requester for a lock on a resource has waited longer than the installation-specified RESOURCE TIMEOUT limit on the DB2 install panel DSNTIPI.

DEADLOCKS
The number of times lock suspension ultimately resulted in a deadlock. This happens when two or more application processes each hold locks on resources that the others need and without which they cannot proceed. A single process accessing data through an unclustered index can sometimes experience a deadlock between a data page and an index page. If deadlocks are not very frequent, they might not impact performance.

ESCAL.(SHARED)
Count of lock escalations to shared mode. Number of times the LOCKS PER TABLE(SPACE) parameter on the panel DSNTIPJ was exceeded and the table space lock was promoted from a page lock (IS) to a table space lock (S) for this thread. Escalation can cause unpredictable response times.

The lock escalation to shared mode should only happen on an exception basis. For example, a REPEATABLE READ application references most pages in a table.

ESCAL.(EXCLUS)
Count of lock escalations to exclusive mode. Number of times the LOCKS PER TABLE(SPACE) parameter on the panel DSNTIPJ was exceeded and the table space lock was promoted from a page lock (IX) to a table space lock (X) for this thread. Escalation can cause unpredictable response times.

MAX PG/ROW LOCKS HELD
Count of the maximum number of page or row locks concurrently held against all table spaces by a single application during its execution. This count cannot exceed the value of the LOCKS PER USER installation parameter value (panel DSNTIPJ). Once the limit is reached, the next attempt to obtain a lock results in a RESOURCE UNAVAILABLE return code, and the SQL request is not processed.

LOCK REQUEST
The number of times a lock on a resource was requested.

CHANGE REQUEST
The number of times a lock change was requested, for example, to promote a shared page lock to exclusive lock.

LOCK SUSPENSIONS
The number of resource conflicts. A suspension is a wait for a lock and each of these waits might contribute adversely to DB2 performance. The suspension can ultimately result in normal resumption, timeout or deadlock. The number of lock suspensions is a function of the lock requests. Lock suspensions (or conflicts) can occur on either LOCK REQUEST or CHANGE REQUEST.

The ratio of suspensions to lock requests is largely application dependent.

LOCKING CONSIDERATIONS

The following aspects should be considered if concurrency is an issue:
- Application design
- IRLM startup procedure options and DB2 installation options
- DDL LOCKSIZE
- BIND parameters.

Refer to *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for detailed information.

Buffer Pool Activity

```
BP16K0               AVERAGE     TOTAL
-------------------  --------  --------
BPOOL HIT RATIO (%)      6.43       N/A
HPOOL HIT RATIO (%)      0.00       N/A
GETPAGES                30.00        60
GETPAGES-FAILED          5.00        30
BUFFER UPDATES           0.00         0
SYNCHRONOUS WRITE        0.00         0
SYNCHRONOUS READ         0.00         0
SEQ. PREFETCH REQS       0.00         0
LIST PREFETCH REQS       0.00         0
DYN. PREFETCH REQS       0.00         0
PAGES READ ASYNCHR.      0.00         0
HPOOL WRITES             0.00         0
HPOOL WRITES-FAILED      0.00         0
PAGES READ ASYN-HPOOL    0.00         0
HPOOL READS              0.00         0
HPOOL READS-FAILED       0.00         0
```

Figure 115. Accounting Report—Buffer Pool Activity

BPOOL HIT RATIO (%)
The total number of GETPAGE operations, minus the number of pages read from DASD (both synchronously and using prefetch), divided by the total number of GETPAGE operations, multiplied by 100.

HPOOL HIT RATIO (%)
This field is a means of measure for I/O avoidance due to the hiperpool. It is the percentage of pages moved into the virtual pool that could be retrieved from the hiperpool instead of reading them from DASD. The field is calculated as follows:

(Pages read from hiperpool / (Pages read from hiperpool + Pages read from DASD)) * 100

GETPAGES
The number of GETPAGE requests. Reducing GETPAGEs helps performance. For this, the database design must be analyzed: if indexes can be added to certain tables, then fewer pages will have to be scanned thereby limiting GETPAGEs.

GETPAGES-FAILED
The number of times that a page requested for a query processed in parallel was unavailable because an I/O was in progress or the page was not found in the buffer pool. The agent does not wait, but control is returned to the agent.

This counter is used only when queries are processed in parallel.

BUFFER UPDATES
A nonzero value indicates either SQL INSERT, UPDATE, DELETE activity, merge scan join, and/or activity on the workfiles because of internal sort. You may want to look at the access path and determine if any sort activity can be minimized or avoided.

SYNCHRONOUS WRITE
Total number of immediate writes for a page. Although an immediate write should be rare, a small nonzero value is always expected. A large value

indicates that the system needs tuning. For more information, refer to “Buffer Pool Activity” on page 241 in statistics.

SYNCHRONOUS READ
The number of synchronous read I/O operations.

SEQ. PREFETCH REQS
The number of times sequential prefetch reads were requested. Table, table space, and nonmatching index scans generally use sequential prefetch.

LIST PREFETCH REQS
The number of times list prefetch reads were requested. List prefetch is always used to access data in multiple index access, and to access data from the inner table during a hybrid join. List prefetch is usually employed with a single index when the index has a cluster ratio lower than 80%. List prefetch is sometimes used on indexes with a high cluster ratio if the amount of data estimated to be accessed is too small to make sequential prefetch efficient, but large enough that more than one synchronous read would be needed.

DYN. PREFETCH REQS
The number of times dynamic prefetch reads were requested. Dynamic prefetch is typically used for a SELECT or UPDATE that is run repeatedly, accessing the index for each access.

PAGES READ ASYNCHR.
The number of pages read asynchronously.

Group Buffer Pool Block

```
GROUP BP8K0           AVERAGE     TOTAL
-------------------  --------  --------
READ(XI)-DATA RETUR      0.00         0
READ(XI)-NO DATA RT      1.50         3
READ(NF)-DATA RETUR      1.00         2
READ(NF)-NO DATA RT      2.00         4
PREFETCH PAGES READ      0.00         0
CLEAN PAGES WRITTEN      0.00         0
CHANGED PAGES WRTN       6.00        12
UNREGISTER PAGE          0.00         0
EXPLICIT X-INVALID       0.00         0
WRITE TO SEC-GBP         0.00         0
COMPL CHECKS SUSP        0.00         0
```

Figure 116. Accounting Report—Group Buffer Pool Block

READ(XI)-DATA RETUR
The number of coupling facility read requests required because the buffer was marked invalid. Data is returned from the group buffer pool.

READ(XI)-NO DATA RT
The number of synchronous coupling facility read requests required because the buffer was marked invalid. Data is not returned from the group buffer pool.

READ(NF)-DATA RETUR
The number of coupling facility read requests necessary because the requested page was not found in the buffer pool. Data is returned from the coupling facility.

READ(NF)-NO DATA RT
The number of synchronous coupling facility read requests necessary because the requested page was not found in the buffer pool. Data is not returned from the coupling facility.

PREFETCH PAGES READ
The number of pages read from the group buffer pool due to prefetch under the control of the agent.

CLEAN PAGES WRITTEN
The number of clean pages written to the group buffer pool.

CHANGED PAGES WRTN
The number of changed pages written to the group buffer pool.

UNREGISTER PAGE
The number of coupling facility requests to unregister a page.

EXPLICIT X-INVALID
The number of explicit cross-invalidations.

WRITE TO SEC-GBP
The number of changed pages written to the secondary GBP for duplexing.

COMPL CHECKS SUSP
The number of completion checks for writes to the secondary GBP that were suspended because the write operation was not yet completed.

SQL Activity

```
SQL DML   AVERAGE    TOTAL    SQL DCL          TOTAL    SQL DDL    CREATE   DROP  ALTER
-------- -------- --------    -------------- --------    ---------- ------ ------ ------
SELECT       0.00        0    LOCK TABLE           0    TABLE           1      0      0
INSERT     200.50      401    GRANT                0    TEMP TABLE      0    N/A    N/A
UPDATE       0.00        0    REVOKE               0    AUX TABLE       0    N/A    N/A
DELETE       0.00        0    SET CURR.SQLID       0    INDEX           1      0      0
                              SET HOST VAR.        0    TABLESPACE      1      1      0
DESCRIBE   206.00      412    SET CUR.DEGREE       1    DATABASE        0      0      0
DESC.TBL     0.00        0    SET RULES            0    STOGROUP        0      0      0
PREPARE    206.00      412    SET CURR.PATH        2    SYNONYM         0      0    N/A
OPEN         0.50        1    SET CURR.PREC        0    VIEW            0      0    N/A
FETCH        1.00        2    CONNECT TYPE 1       0    ALIAS           0      0    N/A
CLOSE        0.50        1    CONNECT TYPE 2       0    PACKAGE       N/A      0    N/A
                              SET CONNECTION       0    PROCEDURE       0      0    N/A
                              RELEASE              0    FUNCTION        0      0    N/A
DML-ALL    614.50     1229    CALL                 0    TRIGGER         0      0    N/A
                              ASSOC LOCATORS       0    DIST TYPE       0      0    N/A
                              ALLOC CURSOR         0
                              HOLD LOCATOR         0    TOTAL           3      1      0
                              FREE LOCATOR         0    RENAME TBL      0
                              DCL-ALL              3    COMMENT ON      0
                                                        LABEL ON        0
```

Figure 117. Accounting Report—SQL Activity

If you are looking at a transaction and there are nonzero values for DESCRIBE, DESC.TBL, or PREPARE, it is an indication the transaction is involved in dynamic SQL activity. Generally this is not to be expected. You may want to investigate further.

If you are looking at a transaction, you would generally not expect to see any SQL DCL or SQL DDL activity. However, should there be any SQL DDL activity, make sure to use frequent commits to minimize contentions.

Refer to *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for detailed information.

RID List Processing

```
RID LIST             AVERAGE     TOTAL
-------------------  --------  --------
USED                     0.00         0
FAIL-NO STORAGE          0.00         0
FAIL-LIMIT EXCEEDED      0.00         0
```

Figure 118. Accounting Report—RID List Activity

USED A nonzero value in this field indicates that DB2 has used list prefetch activity. If you are looking at a transaction and list prefetch is used, you may want to look into the access path selection.

FAIL-NO STORAGE
The number of times RID list processing was terminated due to insufficient storage.

This failure occurs when 2 GB limit is reached. If this happens, contact your system administrator.

FAIL-LIMIT EXCEEDED
The number of times RID list processing was terminated due to one or more internal limits exceeded.

The cause of the failure must be investigated, either from statistics record or from performance trace, before increasing the RID list storage size. Without proper investigation, merely increasing the RID list storage size might not solve the problem.

Service Unit Block

```
AVERAGE SU            CLASS 1           CLASS 2
------------   --------------    --------------
CPU                  77505.67          77449.67
 AGENT               18000.33          17944.33
  NONNESTED          18000.33          17944.33
  STORED PRC             0.00              0.00
  UDF                    0.00              0.00
  TRIGGER                0.00              0.00
 PAR.TASKS           59505.33          59505.33
```

Figure 119. Accounting Report—Service Unit Block

The service unit block is provided so you can compare TCB times in a heterogeneous environment. Especially when you are analyzing group-scope reports you should pay attention to the service unit block if tasks were executed on several processors of different speed.

Stored Procedures

```
STORED PROCEDURES   AVERAGE     TOTAL
-----------------  --------  --------
CALL STATEMENTS        0.00         0
ABENDED                0.00         0
TIMED OUT              0.00         0
REJECTED               0.00         0
```

Figure 120. Accounting Report—Stored Procedures

CALL STATEMENTS
The number of SQL CALL statements executed.

ABENDED
The number of times a stored procedure terminated abnormally.

TIMED OUT
The number of times an SQL CALL statement timed out waiting to be scheduled.

REJECTED
The number of times an SQL CALL statement was rejected due to the procedure being in the STOP ACTION(REJECT) state.

Triggers Block

```
TRIGGERS            AVERAGE     TOTAL
-----------------  --------  --------
STATEMENT TRIGGER     15.00        60
ROW TRIGGER            8.00        24
SQL ERROR OCCUR        0.00         0
```

Figure 121. Accounting Report—Triggers Block

STATEMENT TRIGGER
The number of times a statement trigger was activated.

ROW TRIGGER
The number of times a row trigger was activated.

SQL ERROR OCCUR
The number of times an SQL error occurred during the execution of a trigger action.

UDF Block

```
UDF          AVERAGE     TOTAL
---------  --------  --------
EXECUTED       0.00         0
ABENDED        0.00         0
TIMED OUT      0.00         0
REJECTED       0.00         0
```

Figure 122. Accounting Report—UDF Block

EXECUTED
The number of user-defined functions executed.

ABENDED
The number of times a user-defined function abended.

TIMED OUT
The number of times a user-defined function timed out while waiting to be scheduled.

REJECTED
The number of times a user-defined function was rejected.

Logging Block

```
LOGGING               AVERAGE     TOTAL
-------------------  --------  --------
LOG RECORDS WRITTEN      0.00         0
TOT BYTES WRITTEN        0.00         0
```

Figure 123. Accounting Report—Large Objects Block

LOG RECORDS WRITTEN
The number of log records written.

TOT BYTES WRITTEN
The total number of log record bytes written.

ROWID Block

```
ROWID            AVERAGE     TOTAL
-------------  --------  --------
DIRECT ACCESS      0.00         0
INDEX USED         0.00         0
TS SCAN USED       0.00         0
```

Figure 124. Accounting Report—ROWID Block

DIRECT ACCESS
The number of times direct row access was successful.

INDEX USED
The number of times an index was used to locate a record when the ROWID column was specified in the predicate.

TS SCAN USED
The number of times a table or table space scan was used to locate a record when the ROWID column was specified in the predicate.

Optimization Block

```
OPTIMIZATION        AVERAGE     TOTAL
------------------  --------  --------
REOPTIMIZATION         4.00        12
PREP_STMT_MATCH        2.00         4
PREP_STMT_NO_MATCH     4.00         8
IMPLICIT_PREPARES      2.00         4
PREP_FROM_CACHE        1.50         3
CACHE_LIMIT_EXCEED     3.50         7
PREP_STMT_PURGED       2.00         4
```

Figure 125. Accounting Report—Optimization Block

The first field in this block indicates the number of times reoptimization has occurred for this thread. The other six fields describe the caching of prepared SQL statements.

Miscellaneous Block

```
MISCELLANEOUS        AVERAGE     TOTAL
-------------------  --------  --------
MAX STOR LOB VALUES      0.00         0
```

Figure 126. Accounting Report—Miscellaneous Block

MAX STOR LOB VALUES
Maximum storage used for LOB values.

Resource Limit Facility

This section displays the activity in relation to the resource limit facility for the authorization ID or plan. It shows if the plan is executing dynamic SQL statements that are limited by specific service units in the RLF, with the CPU seconds and the highest CPU seconds shown for the authorization ID or plan.

```
RESOURCE LIMIT TYPE             #OCCUR   AVERAGE CPU SECONDS   HIGHEST CPU SECONDS
----------------------------    ------   -------------------   -------------------
INFINITE LIMIT                       1              5.000000              7.000000
```

Figure 127. Accounting Report—Resource Limit Facility

If, for a specific entry, the AVERAGE CPU SECONDS appears close to the SERV.UNITS value (this value is converted to CPU seconds based on the CPU model used to execute dynamic SQL), and the HIGHEST CPU SEC is equal to or higher than the SERV.UNITS value, then the SQL activity report could be useful in determining the actual SQL statements that were exceeding the RLF limit.

Refer to *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for detailed information regarding Resource Limit Facility.

Query Parallelism

```
QUERY PARALLELISM             AVERAGE     TOTAL
---------------------------  --------  --------
MAXIMUM MEMBERS USED              N/A         0
MAXIMUM DEGREE                    N/A         0
GROUPS EXECUTED                  0.00         0
 RAN AS PLANNED                  0.00         0
 RAN REDUCED                     0.00         0
 ONE DB2-COORDINATOR = NO        0.00         0
 ONE DB2-ISOLATION LEVEL         0.00         0
 SEQUENTIAL-CURSOR               0.00         0
 SEQUENTIAL-NO ESA SORT          0.00         0
 SEQUENTIAL-NO BUFFER            0.00         0
 SEQUENTIAL-ENCLAVE SERVICES     0.00         0
MEMBER SKIPPED(%)                0.00       N/A
DISABLED BY RLF                  1.00        13
REFORM PARAL-CONFIG              0.00         0
REFORM PARAL-NO BUF              0.00         0
```

Figure 128. Accounting Report—Query Parallelism

MAXIMUM MEMBERS USED
The largest number of DB2 members that participated in processing queries. This number includes the originating member plus all assisting members. This value applies only to Sysplex parallel processing.

MAXIMUM DEGREE
Maximum degree of parallelism executed among all the parallel groups. This field indicates the extent to which query parallelism applies.

The degree can be set on the BIND and REBIND commands or the SET CURRENT DEGREE statement. The default is 1, so no parallelism is used unless requested. If a query is I/O bound, I/O parallelism can help to reduce the response time. If a query is CPU bound, query CP and Sysplex parallel processing can reduce the response time. If both conditions apply, the benefits for the query can be very large.

GROUPS EXECUTED
Total number of parallel groups that have been executed.

RAN AS PLANNED
Total number of parallel groups which executed to the planned parallel degree. This field is incremented by one for each parallel group that executed with the planned degree of parallelism (as determined by DB2).

RAN REDUCED
Total number of parallel groups that processed to a parallel degree less than planned because of a storage shortage or contention on the buffer pool. If this field is not zero, increase the size of the current buffer pool by using the ALTER BUFFERPOOL command, or use the ALTER TABLESPACE statement to assign table spaces accessed by this query to a different buffer pool.

ONE DB2-COORDINATOR = NO
Total number of parallel groups executed on a single DB2 due to the Coordinator subsystem value being set to NO.

ONE DB2-ISOLATION LEVEL
Total number of parallel groups executed on a single DB2 due to repeatable-read isolation.

SEQUENTIAL-CURSOR
Total number of parallel groups that fall back to sequential operation because of a cursor that can be used for update or delete.

SEQUENTIAL-NO ESA SORT
Total number of parallel groups that fall back to sequential operation because of a lack of MVS/ESA sort support.

SEQUENTIAL-NO BUFFER
Total number of parallel groups that fall back to sequential mode due to a storage shortage or contention on the buffer pool.

SEQUENTIAL-ENCLAVE SERVICES
Total number of parallel groups executed in sequential mode due to MVS/ESA Enclave Services unavailability.

MEMBER SKIPPED(%)
The percentage of Sysplex parallel groups that were not distributed as planned. This value is calculated as follows: Number of times a DB2 is bypassed (due to insufficient buffer pool storage) divided by the number of parallel groups (that were intended to run across the data sharing group) multiplied by 100.

The purpose of this field is to indicate situations with insufficient buffers on a member so that the parallelism coordinator has to bypass a DB2 when distributing tasks.

DISABLED BY RLF
Total number of threads where at least one query parallelism was disabled by the Resource Limit Facility, which controls the execution of dynamic SQL statements.

REFORM PARAL-CONFIG
Total number of parallel groups in which DB2 reformulated the parallel portion of the access path due to one of the following:

- A change in the number of active DB2 members
- A change in the processor models on which they run from bind time to run time.

This counter is incremented only on the parallelism coordinator at run time.

REFORM PARAL-NO BUF
Total number of parallel groups in which DB2 reformulated the parallel portion of the access path due to insufficient buffer pool resources. This counter is incremented only on the parallelism coordinator at run time.

Query Parallelism Considerations

Query parallelism is suited to both *data-intensive* and *complex* queries. A query is *data-intensive* if it has a high ratio of I/O compared to CPU and *complex* if it requires a high use of CPU to resolve complex predicates and functions.

Generally, queries are suited to query parallelism when the following apply:

- Partitioned tablespace
- Fast n-way processor
- Adequate buffer space to cater for multiple parallel tasks
- Dedicated environment.

Query I/O parallelism works best when there is:

- High ratio of I/O time to CPU time
  - Large record size
  - Very large tables
  - Query which scans many rows but returns few
  - Tablespace scan rather than index scan.

Query CP and Sysplex parallel processing works best when there is:

- High ratio of CPU time to I/O time
  - Complex and/or many predicates
  - Complex many level subqueries
  - Column and scalar functions
  - Multi-way joins
  - Grouping and ordering clauses
  - Type-2 Index access and tablespace scans.

Performance tuning suggestions:

- Partition data and put partitioned data sets on separate DASD volumes.
- Provide sufficient buffer pool size to maximize parallel degree.
- Try to partition data sets evenly.

  For I/O parallelism, the formula is: Maximum degree of parallelism = (total data size) / (maximum partition size).

  For CP and Sysplex parallel processing, the maximum degree can be greater than the number of partitions.

  Actual degree is limited by an estimated ratio of I/O time to CPU time.
- Run RUNSTATS utility to get partition level statistics.

DDF Activity

```
---- DISTRIBUTED ACTIVITY -----------------------------------------------------------------------------------------------------------
SERVER            : STLEC1           CONVERSATIONS INITIATED:    1.00    TRANSACT.SENT:    1.00    MESSAGES SENT    :      3.00
PRODUCT ID        : DB2              #CONVERSATIONS QUEUED  :       0    #COMMT(1)SENT:       0    MESSAGES RECEIVED:      3.00
METHOD            : DB2 PRIV         SUCCESSFULLY ALLOC.CONV:    1.00    #ROLLB(1)SENT:       0    BYTES SENT       :   1314.00
REQUESTER ELAP.TIME:       10.776739  CONVERSATION TERMINATED:    0.00    SQL SENT     :    2.00    BYTES RECEIVED   :   2076.00
SERVER ELAPSED TIME:        2.952933  MAX OPEN CONVERSATIONS :       1    ROWS RECEIVED:   20.00    BLOCKS RECEIVED  :      1.00
SERVER CPU TIME    :        0.014974  #CONT->LIM.BL.FTCH SWCH:       0    MSG.IN BUFFER:   20.00    STMT BOUND AT SER:      0.00
#DDF ACCESSES      :               1
#COMMIT(2) SENT    :               0  #COMMIT(2) RESP.RECV.  :       0    #PREPARE SENT:       1    #FORGET RECEIVED :         1
#BACKOUT(2) SENT   :               0  #BACKOUT(2) RESP.RECV. :       0    #LASTAGN.SENT:       0
```

Figure 129. Accounting Report—Server DDF Block, Present in a Requester Accounting Record

#CONVERSATIONS QUEUED
The number of conversation requests queued by the Distributed Data Facility waiting for allocation.

When this value is high, you may want to increase the number of conversations by tuning VTAM.

MESSAGES RECEIVED
This field shows the count of the messages the REQUESTER location received from the SERVER location.

More messages might be sent from the server location than are received by the requester location due to the manner in which distributed SQL statements are processed internally.

Ratio of ROWS RECEIVED to MESSAGES RECEIVED might indicate block fetch (isolated SQL assumed). For more information about block fetch, refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide*.

REQUESTER ELAP.TIME
This field shows the amount of elapsed time spent at the requester between the sending of the SQL statement and the receipt of the answer from the server. This includes the processing time in DB2, VTAM, and the network.

If this value is large, it could indicate that block fetch has not been used. To confirm if block fetch is used or not, look at the fields MSG.IN BUFFER and BLOCKS RECEIVED. These fields have nonzero values if block fetch is used.

If this time is much larger than SERVER ELAPSED TIME, there might perhaps be problems in the network. This comparison is not meaningful for DRDA protocol, only for DB2 private protocol.

SERVER ELAPSED TIME
This field shows the amount of elapsed time spent at the server between the actual receipt of the SQL statement and until the answer is sent to VTAM.

This is not applicable to DRDA protocol, only to DB2 private protocol.

Comparison of this time with ELAPSED TIME (CLASS 2) reveals how much time is spent by the allied distributed thread in remote processing.

ROWS RECEIVED
This field shows the count of DB2 rows the REQUESTER location received from the SERVER location.

If the number of ROWS RECEIVED is less than the number of ROWS SENT, it is an indication that the user fetched fewer rows than the server sent. This can happen if block fetch is used.

BLOCKS RECEIVED
This field has a nonzero value if block fetch is used.

SERVER CPU TIME
This field shows the amount of CPU time spent at the server between the actual receipt of the SQL statement and until the answer is sent to VTAM.

This is not applicable to DRDA protocol, only to DB2 private protocol.

If this time is much lower than SERVER ELAPSED TIME, it is an indication of long waits at the server for I/O and other DB2 resources.

Class 1 includes everything in class 2. In block fetch, class 1 and class 2 are approximately equal because the SQL processing is repeated multiple times for a single distributed request. If block fetch is not used, class 1 is typically higher than class 2 because the SQL processing is executed a single time for a single distributed request.

#CONT->LIM.BL.FTCH SWCH
The number of times a switch was made from continuous to limited block fetch mode. This is not applicable to DRDA protocol, only to DB2 private protocol.

When this value is high, you may want to consider tuning VTAM.

MSG.IN BUFFER
This field has a nonzero value if block fetch is used.

Refer to *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for detailed information regarding use of block fetch for improved performance.

```
---- DISTRIBUTED ACTIVITY -------------------------------------------------------------------------------------------
REQUESTER          : D31F              TRANSACTIONS RECV. :    1.00    MESSAGES SENT     :   2561.00    MSG.IN BUFFER:    0.00
PRODUCT ID         : DB2 3.1.0         #COMMIT(1) RECEIVED:       0    MESSAGES RECEIVED:   3585.00    ROWS SENT    :  512.00
METHOD             : DRDA              #ROLLBK(1) RECEIVED:       0    BYTES SENT       : 213651.00    BLOCKS SENT  :    0.00
CONVERSAT.INITIATED:             1.00    SQL RECEIVED       : 2048.00    BYTES RECEIVED   : 310686.00    #DDF ACCESSES:       1
#COMMIT(2) RECEIVED:              512    #COMMIT(2) RES.SENT:     512    #PREPARE RECEIVED:       512    #FORGET SENT :       0
#BCKOUT(2) RECEIVED:                0    #BACKOUT(2)RES.SENT:       0    #LAST AGENT RECV.:         1
#COMMIT(2) PERFORM.:             1023    #BACKOUT(2)PERFORM.:       0    #THREADS INDOUBT :         0
```

Figure 130. Accounting Report—Requester DDF Block, Present in a Server Accounting Record

ROWS SENT
This field shows the count of rows sent from the SERVER location to the REQUESTER location.

If the number of ROWS SENT is more than the number of ROWS RECEIVED, it is an indication that the user fetched fewer rows than the server sent. This can happen if block fetch is used.

Refer to *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for detailed information regarding Limited Block Fetch and Continuous Block Fetch.

Data Sharing

```
DATA SHARING         AVERAGE     TOTAL
------------------  --------  --------
GLOBAL CONT RATE(%)     0.00       N/A
FALSE CONT RATE(%)      0.00       N/A
L-LOCKS XES RATE(%)     0.00       N/A
LOCK REQ   - PLOCKS     0.00         0
UNLOCK REQ - PLOCKS     0.00         0
CHANGE REQ - PLOCKS     0.00         0
LOCK REQ   - XES        0.00         0
UNLOCK REQ - XES        0.00         0
CHANGE REQ - XES        0.00         0
SUSPENDS   - IRLM       0.00         0
SUSPENDS   - XES        0.00         0
SUSPENDS   - FALSE      0.00         0
INCOMPATIBLE LOCKS      0.00         0
NOTIFY MSGS SENT        0.00         0
```

Figure 131. Accounting Report—Data Sharing

GLOBAL CONT RATE(%)
The total number of suspends because of contention, divided by the total number of requests that went to XES (excluding asynchronous requests), multiplied by 100.

Aim for a total global lock contention of less than 5%, preferably less than 2%.

FALSE CONT RATE(%)
The number of false contentions, divided by the total number of contentions, multiplied by 100. A false contention is where two different locks on different resources hash to the same lock entry.

Try to keep the false contention rate below 50% of the total global lock contention.

L-LOCKS XES RATE(%)
The number of Transaction Locks (L-Locks) that have been propagated to XES and the Coupling Facility, expressed as a percentage of lock requests. This value does not include requests that are suspended.

LOCK REQ - PLOCKS
The number of lock requests for P-locks.

UNLOCK REQ - PLOCKS
The number of unlock requests for P-locks.

CHANGE REQ - PLOCKS
The number of change requests for P-locks.

LOCK REQ - XES
The number of logical and physical lock requests propagated to MVS XES.

UNLOCK REQ - XES
The number of logical and physical unlock requests propagated to MVS XES.

CHANGE REQ - XES
The number of logical and physical change requests propagated to MVS XES.

SUSPENDS - IRLM
The number of suspensions due to IRLM global resource contention (IRLM lock states were in conflict).

SUSPENDS - XES
The number of suspensions due to MVS XES global resource contention (MVS XES lock states were in conflict but IRLM lock states were not).

SUSPENDS - FALSE
The number of suspensions due to false contention. This happens when different resource names hash to the same entry in the coupling facility lock table. This causes MVS XES to detect contention on the hash class; however, when MVS XES determines that there is no real conflict on the resource, the contention is called *false*.

Make sure that this counter reflects a low percentage of the total global lock contention, that is, of the sum of the three *SUSPENDS* fields. False contention should not exceed 50% of total global lock contention.

INCOMPATIBLE LOCKS
The number of global lock or change requests denied or suspended due to an incompatible retained lock.

NOTIFY MSGS SENT
The number of notify messages sent.

Chapter 17. Monitoring the Access Path—Explain

Tuning DB2

This chapter identifies and describes the specific DB2 data which is reported for the purpose of tuning DB2. For general tuning advice on DB2, refer to the DB2 Administration Guide 'Performance, Monitoring, and Tuning' chapters for the specific release of DB2.

Use the explain report set and source explain to find out the access path selected by DB2 for a given SQL statement. This information is based on the SQL EXPLAIN statement and the relevant DB2 catalog information.

The explain information is useful in problem determination and application design and tuning.

For information on how to produce explain reports, refer to "Explain" on page 68 and for information on how to use source explain, refer to "Source Explain" on page 71.

The explain reports can be based on SQL statements found in:
- An application plan
- An application package
- A saved QMF query (SQL format)
- A specified SQL statement
- A specified query number.

Use the source explain to explain SQL statements that are embedded in a source program or SPUFI input.

This chapter focuses on explaining SQL statements in plans and packages.

The explain report set can show information for packages bound at a remote location. If a list of plans to be explained contains a remotely bound package, DB2 PM EXPLAIN automatically connects to the server and explains the remote package. Alternatively, you can specify the server location to which DB2 PM EXPLAIN is to connect and the plans and packages you want explained.

Explaining Statements in a Plan or in a Package

Product-Sensitive Programming Interface

When the statements of a plan or a package are being explained, the execution time can be relatively long if the plan or package holds many explainable SQL statements, because the explain function selects data from some nonindexable DB2 system catalog tables. To improve performance, refer to "Speeding Up Your Reports" on page 223.

Explain

To get a quick idea of the state of the SQL statements in a large plan or package, produce a summary report. The following example shows you how:

```
EXPLAIN .... LEVEL(SUMMARY)
```

The report created shows the access path for each SQL statement in one line for each plan.

```
REPORT ON:10/21/98 13:58:04          DB2 PM (V6)               SUMMARY PAGE
                                       EXPLAIN
                                    SUMMARY REPORT        USER AUTHID: USR1



THE FOLLOWING   1 DB2 PM EXPLAIN REQUESTS WERE PROCESSED:             PAGE NO



  1:    APC6 PLAN       LOXXPLAN
             SUMMARY REPORT REQUESTED
      LOXXPLAN     67  P L  MULTIPLE INDEX ACCESS PATH - LIST PREFETCH
      LOXXPLAN     67  P    MATCHING INDEX SCAN(1/1)-INDEX ONLY
      LOXXPLAN     67  P    MATCHING INDEX SCAN(1/1)-INDEX ONLY
      LOXXPLAN     67  P    MATCHING INDEX SCAN(1/1)-INDEX ONLY
      LOXXPLAN     67  P    MATCHING INDEX SCAN(1/1)-INDEX ONLY
      LOXXPLAN     67  P    ADDITIONAL SORT FOR ORDER BY
      LOXXPLAN     67  P    MATCHING INDEX SCAN(1/1)-DATA PAGES
      LOXXPLAN     71  P S  TABLE SPACE SCAN-NO INDEX WILL BE USED
      LOXXPLAN     71  P    ADDITIONAL SORT FOR ORDER BY



DB2 PM EXPLAIN PROCESSING COMPLETED.
```

Figure 132. Explain Summary Report

As the example shows, a *table space scan* will be performed in the first plan of statement number 71. If you would like to know the reason for the table space scan, but would still like to have a fast DB2 PM explain execution time, the next step is to generate a report where all the explain information and access path information is produced, but where information on indexes, tables, and table spaces is excluded. Such a report will be generated using the following statement:

```
EXPLAIN .... LEVEL(BASIC)
```

You could also limit the report to show information only for the statement you are interested in by specifying the name of the plan and the statement number range as follows:

```
EXPLAIN
PLAN  (LOXXPLAN)
LEVEL (BASIC)
FIRST (71)
LAST  (71)
```

This statement will generate a report similar to Figure 133 on page 219, where only the explain data and the actual access path is shown.

```
ACTUAL AT:12/10/98 16:26:49      DB2PM (V6)                PAGE       :     1-1
                                 EXPLAIN PLAN              DB2 RELEASE: V6
LOCATION: SYSDSN5                 LOXXPLAN                 USER AUTHID: PMDEV
SUBSYSTEM:SG51                      BASIC                  CURR.SQLID : PMDEV


PLAN LOCATION     :SYSDSN5
PLAN NAME         :LOXXPLAN
PACKAGE LOCATION  :SYSDSN5
PACKAGE COLLECTION:LOXXPLANCOL
PACKAGE ID        :LOXXPLAN
PACKAGE VERSION ID:DFLT
STATEMENT NUMBER  :      71

SQL STATEMENT READ FROM SYSIBM.SYSPACKSTMT:

 DECLARE CURSOR5 CURSOR FOR
 SELECT *
 FROM DSN8610.DEPT
 WHERE ((DEPTNO = 'D11' AND MGRNO = '000060') OR ADMRDEPT = 'D11')
 ORDER BY DEPTNAME¦

STATUS   : COMPILED-BOUND USING DEFAULTS FOR INPUT VARIABLES
ISOLATION: CURSOR STABILITY

-------------------------------------------------------------------------------
DSN_STATEMENT_TABLE
EXPLAIN TIME: 1998-12-10-16.19.31.157642
QUERY NO:       71, APPL.NAME   :
                    PROGNAME    : LOXXPLAN
                    COLLID      : LOXXPLANCOL
                    GROUP MEMBER: SG51
                    STMT TYPE   : SELECT

COST CATEGORY:  A
REASON:
ESTIMATED PROCESSOR COST (MS):            3
          PROCESSOR COST (SU):            5

-------------------------------------------------------------------------------
PLAN TABLE DATA    OWNER: PMDEV
BIND TIME : 1998-12-10-16.19.31.157642

QUERYNO:      71, ACC. TYPE:   R, PLAN NO:   1, TABLE NAME: DEPT
Q BLOCK NO.:   1, MATCHCOLS:   0, TAB. NO:   1,   -  OWNER: DSN8610
DATE: 1998-12-10, PLAN:         , METHOD :   0, INDEX NAME:
TIME: 16:19:31.1, PCKG: LOXXPLAN, IDXONLY : NO,   -  OWNER:
TS LOCKMODE:  IS, COL.FUNC. :   , PREFETCH:  S, 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.: SG51
COLLECTION : LOXXPLANCOL        , VERSION    : DFLT

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 133. Explain Report (Part 1 of 5)

```
ACTUAL AT:12/10/98 16:26:49      DB2PM (V6)               PAGE       :     1-2
                                 EXPLAIN PLAN             DB2 RELEASE: V6
LOCATION: SYSDSN5                 LOXXPLAN                USER AUTHID: PMDEV
SUBSYSTEM:SG51                      BASIC                 CURR.SQLID : PMDEV


THE ACCESS PATH CHOSEN BY DB2 AT 16:19:31.1   ON 1998-12-10

+-------------------------------------------------------------------+
! TABLE SPACE SCAN - NO INDEX WILL BE USED                          !
! STANDARD SEQUENTIAL PREFETCH WILL BE PERFORMED                    !
! LOCK MODE IS SHARE LOCK FOR THE PAGE                              !
! PAGE RANGE SCAN WILL NOT BE USED                                  !
!                                                                   !
!                                                                   !
+-------------------------------------------------------------------+
===============================================================================
PLAN NAME:    LOXXPLAN     LOCATION:    SYSDSN5                   CHANGES

CREATOR  :    PMDEV        PMDEV
BIND DATE:    1998-12-10   1998-12-10
BIND TIME:    16:19:31.39  16:09:49.77                              <===
BOUND BY :    JEN          JEN
QUALIFIER:    PMDEV        PMDEV
BASE SIZE:        2040         2040
AVG. SIZE:           0            0
CACHESIZE:        1024         1024
PLENTRIES:           1            1
SYS.ENTR.:           0            0
SQL STMTS:           7            7
VALIDATE :    BIND         BIND
ISOLATION:    CUR.STAB.    CUR.STAB.
VALID    :    YES          YES
OPERATIVE:    YES          YES
ACQUIRE  :    USE          USE
RELEASE  :    COMMIT       COMMIT
DEFERPREP:    NO           NO
CURR.SERV:    N/P
                           N/P
DEGREE   :    1            1
REOPTIM. :    NO           NO
DYN.RULES:
KEEP DYN.:    NO           NO
SQLRULES :    DB2          DB2
PATH     :
DDF PROT.:    DRDA         DRDA
FNCT.RES.:    1998-12-10-161998-12-10-16
DISCONNCT:    EXPLICIT     EXPLICIT
OPTHINTID:
STORED BY:    PMDEV        PMDEV
STORED AT:    1998-12-10   1998-12-10

===============================================================================
DBRM/PACK PC-DATE    PC-TIME  LANG. SQLSTMT  CHARSET   COMMA DEC31 TYPE  REL.
--------- -------    -------  ----- -------  -------   ----- ----- ----  ----
```

Figure 133. Explain Report (Part 2 of 5)

```
ACTUAL AT:12/10/98 16:26:49      DB2PM (V6)               PAGE       :     1-3
                                 EXPLAIN PLAN             DB2 RELEASE: V6
LOCATION: SYSDSN5                 LOXXPLAN                USER AUTHID: PMDEV
SUBSYSTEM:SG51                      BASIC                 CURR.SQLID : PMDEV


PLAN LOCATION     :SYSDSN5
PLAN NAME         :LOXXPLAN
PACKAGE LOCATION  :SYSDSN5
PACKAGE COLLECTION:LOXXPLANCOL
PACKAGE ID        :LOXXPLAN
PACKAGE VERSION ID:DFLT
STATEMENT NUMBER  :      71

SQL STATEMENT READ FROM SYSIBM.SYSPACKSTMT:

 DECLARE CURSOR5 CURSOR FOR
 SELECT *
 FROM DSN8610.DEPT
 WHERE ((DEPTNO = 'D11' AND MGRNO = '000060') OR ADMRDEPT = 'D11')
 ORDER BY DEPTNAME

STATUS   : COMPILED-BOUND USING DEFAULTS FOR INPUT VARIABLES
ISOLATION: CURSOR STABILITY

-------------------------------------------------------------------------------
PLAN TABLE DATA    OWNER: PMDEV
BIND TIME : 1998-12-10-16.19.31.157642

QUERYNO:      71, ACC. TYPE:    , PLAN NO:   2, TABLE NAME:
Q BLOCK NO.:   1, MATCHCOLS:   0, TAB. NO:   0,   -  OWNER:
DATE: 1998-12-10, PLAN:         , METHOD :   3, INDEX NAME:
TIME: 16:19:31.1, PCKG: LOXXPLAN, IDXONLY : NO,   -  OWNER:
TS LOCKMODE:    , 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.: SG51
COLLECTION : LOXXPLANCOL        , VERSION    : DFLT

SORTN TABLE -UNIQUE: N, JOIN: N, ORDER BY: N, GROUP BY: N, PARAL.GROUP:     0
SORTC TABLE -UNIQUE: N, JOIN: N, ORDER BY: Y, GROUP BY: N, PARAL.GROUP:     0
-------------------------------------------------------------------------------

THE ACCESS PATH CHOSEN BY DB2 AT 16:19:31.1   ON 1998-12-10

+-----------------------------------------------------------------+
! ADDITIONAL SORT FOR ORDER BY                                    !
! PAGE RANGE SCAN WILL NOT BE USED                                !
!                                                                 !
!                                                                 !
+-----------------------------------------------------------------+
===============================================================================
```

Figure 133. Explain Report (Part 3 of 5)

```
ACTUAL AT:12/10/98 16:26:49      DB2PM (V6)               PAGE       :     1-4
                                 EXPLAIN PLAN             DB2 RELEASE: V6
LOCATION: SYSDSN5                 LOXXPLAN                USER AUTHID: PMDEV
SUBSYSTEM:SG51                      BASIC                 CURR.SQLID : PMDEV

LOCATION     : SYSDSN5
COLLECTION ID: LOXXPLANCOL
PACKAGE ID   : LOXXPLAN
VERSION ID   : DFLT
CONSIST.TOKEN: X'162F6A8009089EA4'
PDSNAME      : JEN.PARALLEL.COBOL.DBRMLIB

OWNER        : PMDEV                QUOTE        : APOSTROPHE
CREATOR      : JEN                  COMMA        : PERIOD
BIND DATE    : 1998-12-10           HOSTLANG     : VS COBOL II
BIND TIME    : 16.19.31.157642      CHARSET      : ALPHANUMERIC
CREATE DATE  : 0001-01-01           MIXED        : NO
CREATE TIME  : 00.00.00.000000      DEC31        : NO
QUALIFIER    : PMDEV                DATA CURRENCY: INHIBIT BLOCKING
BASE SIZE    :       1576           SQLERROR     : NOPACKAGE
AVERAGE SIZE :       8428           SOURCE       : DBRM
SYSENTRIES   :          0           PRECOMP. DATE: 1998-12-10
SQL STATEMENT:          1           PRECOMP. TIME: 16.19.16.446586
VALIDATE     : RUN                  VALID        : YES
ISOLATION    : CURSOR STABILITY     OPERATIVE    : YES
RELEASE      : CHECK PLAN           REOPTIMIZAT. : NO
DEGREE       : ANY                  DEFERPREPARE : INHERITED FROM PLAN
KEEP DYNAMIC : DRDA                 DDF PROTOCOL : INHERITED FROM PLAN
TYPE OF PACK.: BIND PACKAGE         OPT_HINT_ID  :
FNCT.RESOLVED: 1998-12-10-16.19.31.146517
      ....5....0....5....0....5....0....5....0....5....0....5....0....5..
PATH:
===============================================================================
```

Figure 133. Explain Report (Part 4 of 5)

```
REPORT ON:12/10/98 16:26:49        DB2 PM (V6)                  SUMMARY PAGE
                                      EXPLAIN
                                   SUMMARY REPORT        USER AUTHID: PMDEV


THE FOLLOWING   1 DB2 PM EXPLAIN REQUESTS WERE PROCESSED:               PAGE NO


  1:    SG51 PLAN       LOXXPLAN
             BASIC REPORT REQUESTED
      LOXXPLAN      71 P S  TABLE SPACE SCAN-NO INDEX WILL BE USED         1-2
      LOXXPLAN      71 P    ADDITIONAL SORT FOR ORDER BY                   1-3


DB2PM EXPLAIN PROCESSING COMPLETED.
```

Figure 133. Explain Report (Part 5 of 5)

If you need table and table space information to determine the reasons for the selected access paths, you can now generate a quick explain report without index information, as follows:

```
EXPLAIN .... LEVEL(DETAIL) INDEX(NO)
```

Finally, if you want to generate a report with all available catalog information including key distribution and all index information created on the table, use the following:

```
EXPLAIN .... LEVEL(KEYDIST) INDEX(ALL)
```

End of Product-Sensitive Programming Interface

Speeding Up Your Reports

Product-Sensitive Programming Interface

The time taken, and resources used, to produce DB2 PM Batch explain reports depends directly on the functions and the processing options specified. The number of accesses to the DB2 system catalog is the most important factor. The following catalog tables are nonindexed:
- SYSIBM.SYSSTMT
- SYSIBM.SYSDBRM.

One way to speed things up is to create user-defined indexes on the catalog tables:
- Two indexes for SYSIBM.SYSSTMT: one on the columns PLCREATOR, PLNAME, NAME, and STMTNO and one on the columns PLNAME, SEQNO, STMTNO, and SECTNO
- One index for SYSIBM.SYSDBRM: on the columns PLCREATOR, PLNAME, and NAME.

Note, however, that in addition to the usual costs of indexes (for more information, refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide*), processing can take longer in an environment with frequent DDL and static bind activity.

Another way is to create copies of these catalog tables and add indexes to these copies. A sample member, DGOYCOPY in SDGOSAMP, creates such copies. DB2 does not automatically update the copied tables. To get up-to-date DB2 PM explain information, copy the data from the DB2 catalog tables to your tables on a daily basis. To do this, use DGOYRSQL provided in SDGOSAMP. Modify this sample to suit your installation.

You can also instruct DB2 PM to skip production of index and key information using the LEVEL and INDEX options.

The execution time also depends on the number of rows in the accessed PLAN_TABLE. If the table is large, you can create an index on PROGNAME, QUERYNO, or delete unnecessary rows from your PLAN_TABLE.

For further information, see the *DB2 PM Program Directory*.

End of Product-Sensitive Programming Interface

What to Look for in the EXPLAIN Information

Product-Sensitive Programming Interface

When an SQL statement is being explained, it is very important that explaining takes place on the production DB2 subsystem, or at least on a DB2 subsystem where the catalog statistics have been updated to reflect the real production system, for example, in terms of table size, available indexes, and other key values. Online Monitor explain lets you specify the “current server” on which explain is to be executed. This option makes it possible that you are connected to a DB2 test subsystem while you execute the explain on the remote production system.

Following are some examples of explain information that can be useful in determining why an application does not achieve the expected performance:

- **Access path chosen**

  Table space scans and nonmatching index scans should be avoided unless you intend to access all rows in a given table or the table is very small. If the table has one or more indexes, try to reconstruct the SQL statement in such a way that DB2 chooses a better access path. If there is no index, consider creating one.
- **Index Only Access**

  When selecting a few column values only, you should consider the possibility of including these few columns in the column list of one of the indexes. In this way, all requested data can be found in the index. The access path message informs you if you succeed in doing so. Likewise, if you are selecting a maximum value, you might consider building a descending index on that column (or ascending index, if the minimum value is requested). In this way, you can even avoid scanning of leaf pages in the index structure.
- **Clustering versus clustered**

  Make sure that if a clustering index has been chosen by DB2, the actual index is clustered. On the Index Information window, if the clustered value is `NO`, or if the cluster ratio is less than 95%, the table space might need a reorganization in order to bring the data rows into clustering sequence.
- **Number of matching columns**

  On the Plan Table Data panel, if DB2 has selected a matching index scan, you should verify on the Index Information window that the number of columns used in the index is what you expect.
- **Active pages versus pages with rows**

  Verify that the number of pages with rows is about the same as active pages, especially if you are performing table space scans. In other words, the value shown in the *Percentage of pages used* field on the Table Information window should be as close as possible to `100` percent.
- **Number of tables per tablespace**

  On the Table Space Information window, you should monitor the *Tables* field. This field shows the number of tables located in the tablespace. If the access path is “Tablespace scan” and the tablespace is not segmented, it is recommended that there be only one table in the tablespace. In a nonsegmented tablespace, all tables are scanned, not only the selected table.
- **Host variable definitions versus column definitions**

  An inconsistent definition of host variables shown on the Host Variable Definition window, compared to the corresponding column definitions shown on the Key Column Selection window, can indicate an inefficient access path selection, resulting from a possible disqualification of index usage. If, for example, an index column is defined as 3 characters, and that column is being compared in a WHERE-clause with a host variable defined as 4 characters, then DB2 does not base its access path selection on the mentioned index. You should verify that a column and a host variable being compared in a WHERE-clause have compatible definitions.

End of Product-Sensitive Programming Interface

Chapter 18. Monitoring the Subsystem—Statistics

Tuning DB2

This chapter identifies and describes the specific DB2 data which is reported for the purpose of tuning DB2. For general tuning advice on DB2, refer to the DB2 Administration Guide ‘Performance, Monitoring, and Tuning’ chapters for the specific release of DB2.

This chapter shows how to interpret the key fields in the statistics reports.

In statistics, the layouts of reports and traces are very similar. The difference is that a statistics report summarizes data over one or more user-defined intervals and a trace presents the data as the difference (delta) between the statistics recorded in two consecutive record pairs.

The report examples shown in this section are from the short report.

The fields shown in statistics reports are organized in blocks. You can decide which blocks and which fields within blocks you want to include on your report. For more information, refer to “Tailoring Report Layouts” on page 115.

For information on how to produce statistics reports and examples of statistics reports, refer to “Statistics” on page 78.

Highlights of System Performance

The highlights block shows key information about the system, such as the number of threads and commits during the reporting interval.

```
---- HIGHLIGHTS ----------------------------------------------------------------------------------------------------------------
INTERVAL START: 04/06/99 20:10:37.48    INTERVAL ELAPSED: 27:18.88309    INCREMENTAL BINDS      :     0.00    DBAT QUEUED:      N/P
INTERVAL END  : 04/06/99 20:37:56.36    OUTAGE ELAPSED  :    0.000000    AUTH SUCC.W/OUT CATALOG:    11.00    DB2 COMMAND:    51.00
SAMPLING START: 04/06/99 20:10:37.48    TOTAL THREADS   :       38.00    BUFF.UPDT/PAGES WRITTEN:     4.93    TOTAL API  :     0.00
SAMPLING END  : 04/06/99 20:37:56.36    TOTAL COMMITS   :       63.00    PAGES WRITTEN/WRITE I/O:     1.93    MEMBER     :      N/A
```

Figure 134. Statistics—Highlights

INTERVAL START
The start time of the period represented by data in the interval. This is the FROM time specified by the user.

INTERVAL END
The end of the time period represented by data in the interval. This is the TO time specified by the user.

SAMPLING START
The timestamp at the beginning of the DB2 sampling interval. It is from the first delta record used in the interval specified by the user.

SAMPLING END
The timestamp at the end of the DB2 sampling interval. It is from the last delta record used in the interval specified by the user.

INTERVAL ELAPSED
The report interval elapsed time.

OUTAGE ELAPSED
The elapsed time for the period where statistics were not available.

TOTAL THREADS
The number of successful create thread requests. This does not include DBATs.

TOTAL COMMITS
Total number of commits.

INCREMENTAL BIND
The number of incremental binds executed.

If a plan is bound with VALIDATE(RUN), DB2 performs validity checks (such as authorizations and existence of referenced DB2 objects) at bind time and rechecks any failures at run time. This can result in catalog contention and degraded application performance, depending on the number of statements flagged and the number of times they are executed. Therefore, VALIDATE(RUN) should be avoided as much as possible. Ensure that all objects are created and all privileges are granted before binding, and select the VALIDATE(BIND) option.

In addition to plans bound with VALIDATE(RUN), this counter is incremented for plans using DB2 private protocol.

AUTH SUCC.W/OUT CATALOG
The number of successful authorization checks not using the DB2 catalog, including plan cache checks and public checks.

For transaction level security, ENABLE and DISABLE on BIND PACKAGE should be used to ensure adequate security. Granting execute authority on the plan to public might be quite adequate.

Refer to *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for detailed information.

BUFF.UPDT/PAGES WRITTEN
The number of buffer updates per pages written.

This ratio measures the degree of updating on a per-page basis. It is largely application dependent.

PAGES WRITTEN/WRITE I/O
The number of pages written from the buffer pool to DASD per synchronous or asynchronous write I/O.

Writing multiple pages per call to media manager should be as high as possible. This count does not include preformatting I/O, such as I/O needed to prepare a data set for use.

DBAT QUEUED
The number of times a DBAT was queued because the MAX REMOTE ACTIVE value on the DSNTIPE installation panel was reached.

Monitoring this field is useful in developing the right setting for the DSNTIPE parameter MAX REMOTE ACTIVE. This parameter is used to control the number of DBATs that can be concurrently active in the DB2 system.

Rule of thumb: About 1% DBAT queuing is acceptable.

DB2 COMMAND
The total number of DB2 commands issued, including unrecognized commands.

TOTAL API
The total number of calls made to IFI.

MEMBER
In group-scope reports, this field shows the number of DB2 subsystems for the reported data sharing group.

MAX REMOTE ACTIVE

The MAX REMOTE ACTIVE option on the install panel DSNTIPE is used to specify the number of database access threads that can be active at the same time as opposed to MAX USERS on the same panel, which specifies the maximum number of allied threads. The combined maximum allowed for MAX USERS and MAX REMOTE ACTIVE cannot exceed 2000.

MAX REMOTE CONNECTED on the DB2 install panel DSNTIPE represents the number of database access threads that can concurrently exist. This number cannot exceed 25 000.

The total number of inactive database access threads is the difference between MAX REMOTE CONNECTED and MAX REMOTE ACTIVE.

An installation might choose, via DDF THREADS on the DB2 install panel DSNTIPR, to have database access threads considered inactive when the last operation of the thread was a commit or rollback, all packages used by the database access thread had the RELEASE (COMMIT) option, and the thread holds no database locks (including not having any cursors open with the HOLD option). When this is the case, the thread is removed from active thread lists and moved to inactive thread lists.

If the limit set by MAX REMOTE ACTIVE parameter is reached, remote SQL requests are queued until a DBAT can be created. The number of times queueing occurred is shown by the field DBAT QUEUED. If necessary, the value of MAX REMOTE ACTIVE should be increased.

SQL Activity

Check the SQL information blocks to see the SQL activity in your system.

```
SQL DML   QUANTITY    SQL DCL         QUANTITY    SQL DDL     QUANTITY
--------  --------    --------------  --------    ----------  --------
SELECT        0.00    LOCK TABLE          0.00    CREATES        16.00
INSERT       38.00    GRANT               0.00    DROPS           9.00
UPDATE        9.00    REVOKE              0.00    ALTERS          0.00
DELETE        0.00    SET HOST VAR.       0.00    RENAME TBL      0.00
PREPARE      92.00    SET SQLID           4.00    COMMENT ON      0.00
DESCRIBE     84.00    SET DEGREE          0.00    LABEL ON        0.00
DESC.TBL      0.00    SET RULES           0.00    TOTAL          25.00
OPEN          5.00    SET PATH            0.00
CLOSE         5.00    SET PRECISION       0.00
FETCH        33.00    CONNECT TYPE 1      0.00
TOTAL       266.00    CONNECT TYPE 2      0.00
                      RELEASE             0.00
                      SET CONNECTION      0.00
                      ASSOC LOCATORS      0.00
                      ALLOC CURSOR        0.00
                      HOLD LOCATOR        0.00
                      FREE LOCATOR        0.00
                      TOTAL               4.00
```

Figure 135. Statistics—SQL Activity

A *workload profile* can be defined from the following fields:
- TOTAL DML
- TOTAL DCL
- TOTAL DDL.

If the workload or its profile has changed from previous reports, new applications have been implemented, or there has been an increase in the number of, for example, QMF and SPUFI users.

Subsystem Services

The subsystem services block shows information about number of threads, commits, checkpoints, and thread queuing in the reporting interval.

```
SUBSYSTEM SERVICES                  QUANTITY
--------------------------------  --------
IDENTIFY                             20.00
CREATE THREAD                        38.00
SIGNON                                0.00
TERMINATE                            56.00
ROLLBACK                              0.00
COMMIT PHASE 1                        0.00
COMMIT PHASE 2                        0.00
READ ONLY COMMIT                      0.00
UNITS OF RECOVERY GONE INDOUBT        0.00
UNITS OF RECOVERY INDOUBT RESOLV      0.00
SYNCHS (SINGLE PHASE COMMIT)         63.00
QUEUED AT CREATE THREAD               0.00
SYSTEM EVENT CHECKPOINT               1.00
```

Figure 136. Statistics—Subsystem Services

IDENTIFY
The number of successful connections to DB2 from an allied address space (for example TSO, BATCH, CICS, IMS, CAF, or UTILITY).

CREATE THREAD
The number of threads created. Thread creation can be a significant part of the cost in a short transaction. Thread reuse is beneficial to improving performance.

SIGNON
The number of signons that have occurred in IMS or CICS. If the number of signons is greater than the number of create thread occurrences, some threads have been reused. In the case of the TSO attachment facility and the call attachment facility (CAF), there is no signon because the user is identified when the TSO address space is connected.

THREAD REUSE

The term *thread reuse* only applies to IMS and CICS attachments. In the case of the TSO attachment facility and the call attachment facility (CAF), threads cannot be reused because the threads are allocated to the user address space.

Thread reuse should be considered in the following cases:

- If transaction volume is high:

  High volume transactions should achieve a high percentage of thread reuse. If threads are reused on low volume transactions, the number of threads needed increases because these threads are not automatically terminated by IMS when not being used. This might result in too many idle threads for the level of the DB2 workload. Under CICS, protected threads are terminated after the purge cycle (user-specified since DB2 V5, with a minimum of 30 seconds and up to 59 minutes and 59 seconds) if no transaction eligible to reuse the thread has been received.
- If thread creation cost is significant:

  As a rule of thumb, more than 5% of the total CPU cost of transaction processing is considered significant.

The ACQUIRE and RELEASE parameters of BIND should be specified to minimizethe thread creation cost, while providing the needed concurrency:

- If most of the application plan’s SQL statements are executed, then ACQUIRE(ALLOCATE) is cheaper than ACQUIRE(USE).
- If only a small number of the SQL statements are executed, ACQUIRE(USE) becomes cheaper and improves concurrency, because the required resources are only acquired (locked) when the plan actually references (uses) them. An example would be a generalized plan used by many different transactions. It would contain multiple logic paths referencing different tables.

  Note that when packages are involved, ACQUIRE(USE) is always implicitly used.
- Concurrency in thread reuse is based on page locking provided by the IS and IX intent locks, whose duration is governed by ACQUIRE and RELEASE of BIND.

  RELEASE(DEALLOCATE) is strongly recommended for thread reuse transactions to reduce transaction CPU time.

When thread reuse is implemented, monitor the EDM pool. It must be sufficient to accommodate expanding plans where the next transaction requires additional plan sections over those that are already part of the EDM pool.

TERMINATE
: The number of threads that have been terminated. The value of this field is usually greater than the number of create thread occurrences, because it also includes the termination of connections to DB2 (IDENTIFY) and other internal counts.

ROLLBACK
: The number of times a unit of recovery was rolled back successfully. This number also includes successfully aborted agents being associated with threads that use the RRS attachment facility. Some reasons include:
- Application program abend
- Application rollback request
- Application deadlock on database records
- Application canceled by operator
- Thread abend due to resource shortage.

COMMIT PHASE 1
: The number of successful prepare to commit phase-1 requests in a two-phase commit environment such as CICS or IMS. This number also includes successfully prepared agents being associated with threads that use the RRS attachment facility. This value does not include successful single-phase commits or distributed two-phase commits.

COMMIT PHASE 2
: The number of successful commit phase-2 requests in a two-phase environment such as CICS or IMS. This number also includes successfully committed agents being associated with threads that use the RRS attachment facility. This value does not include successful single-phase commits or distributed two-phase commits. A nonzero value for this field indicates that updates have occurred.

READ ONLY COMMIT
: The number of times read operations were completed in a two-phase commit environment.

SYNCHS (SINGLE PHASE COMMIT)
: The number of commits from TSO, CAF, and UTILITY environments.

QUEUED AT CREATE THREAD
: The number of create thread requests queued. This count does not include DBATs.

Monitoring this field is useful in determining the right setting for the MAX USERS option on the DSNTIPE installation panel. This parameter is used to control the number of threads (excluding DBATs) in the DB2 system.

Rule of thumb: About 1% thread queuing is acceptable.

MAX USERS

MAX USERS on the DB2 install panel DSNTIPE is used to specify the maximum number of allied threads that can be allocated concurrently as opposed to MAX REMOTE ACTIVE (MAX REMOTE ACTIVE on the DB2 install panel DSNTIPE) which represents the specifiable limit of active database access threads. The combined maximum allowed for MAX USERS and MAX REMOTE ACTIVE cannot exceed 2 000.

If the limit set by MAX USERS parameter is reached, allied SQL requests are queued (no indication is returned to the user while waiting) until a thread can be created. The number of times queuing occurred is shown by the field QUEUED AT CREATE THREAD. If the value of MAX USERS is too small, thread queuing might result. If necessary, increase the value of MAX USERS.

Environment-specific control is usually needed when determining the number of threads. A DB2 system can connect at the same time to multiple allied address spaces, including CICS and IMS, and the installation might need to be selective. In order to accomplish this, MAX USERS is accompanied by two additional parameters MAX TSO CONNECT and MAX BATCH CONNECT on the Storage Sizes install panel (DSNTIPE).

1. MAX TSO CONNECT limits the number of TSO foreground connections. Usually these will be QMF and SPUFI users. Also TSO transactions executing in the foreground will be controlled by this parameter. MAX TSO CONNECT will usually be greater than the combined number of foreground threads at any one time.
2. MAX BATCH CONNECT limits the number of batch connections. This would include DB2 utilities, OS/390 batch jobs (using CAF) and TSO batch jobs, for example, running QMF reports in the background. MAX BATCH CONNECT approximates the actual number of background threads.

   MAX BATCH CONNECT does NOT control the number of threads coming from IMS batch processing; these are IMS threads.

MAX TSO CONNECT and MAX BATCH CONNECT apply at CONNECTION rather than at CREATE THREAD time. In assigning a value to MAX TSO CONNECT, the installation should consider that at any one time there will normally be fewer online TSO threads than MAX TSO CONNECT connections. This is installation dependent and probably will vary by time of day.

Note that a QMF user creates a thread per query, and ideally it should be terminated during think time. The following options should be considered:
- RESET DATA command
- BOTTOM subcommand
- F parameter setting
- QMF Governor
- Resource Limit facility (RLF).

MAX USERS continued

Determining the appropriate value for MAX USERS is an iterative process. A suggested methodology is shown here. MAX REMOTE ACTIVE is excluded from this scenario.

For these options use both thread and connection counts and do not specify thread counts for IMS/CICS at all. IMS and CICS threads are whatever is left over after deciding on MAX USERS, MAX TSO CONNECT, and MAX BATCH CONNECT.

- Assume MAX USERS is to be 100.
- The question is what should be the settings of MAX TSO CONNECT and MAX BATCH CONNECT. If we are going to give IMS/CICS threads priority over query, as we are likely to do in a production system, we decide to have 70 threads devoted to IMS/CICS (production) transactions.
- This leaves 30 threads to divide up among query, batch, SPUFI, and others.

  The crux is MAX TSO CONNECT: its value is in CONNECTIONS and, given the way TSO online DB2 applications (QMF or TSO transactions) usually work, this value represents, at any instant in time, considerably fewer actual DB2 threads. So set MAX TSO CONNECT to 75. How many DB2 threads from 75 connected TSO regions does this actually represent?

  An adjustment factor needs to be developed, unique for each installation, that might indicate that a MAX TSO CONNECT of 75 represents an average of 25 actual TSO/QMF concurrent threads. Some experience and analysis will be needed.

  Consider starting performance trace for any of the classes 30, 31, or 32 (reserved for installation usage) and for IFCIDs 73 (End create thread), 82 (Begin identify), and 83 (End identify).

  For example:

  ```
  -START TRACE (PERFM) CLASS(30) IFCID(73,82,83)
  ```

  Look for IDENTIFY corresponding to TSO and for those IDENTIFYs how many threads have been created.

  The trace should only be started on an exceptional basis because of the overhead.

  So at this point we have got 70 threads for IMS/CICS and 25 threads for MAX TSO CONNECT. How about MAX BATCH CONNECT?
- MAX BATCH CONNECT becomes 5 by default, and in this scenario, we will say that probably 5 threads are enough for the occasional utility or other batch jobs needed during the processing day. Batch and utility connections are more or less equivalent to a thread.

So in summary: 100 = 70 + 25 + 5

SYSTEM EVENT CHECKPOINT

The value in this field indicates the number of checkpoints taken since DB2 start.

Rule of thumb: In a production environment DB2 should take checkpoints every 10 minutes or so.

The default value for LOGLOAD is 50 000 in DB2 Version 3. The checkpoint frequency is a trade-off between the performance efficiency of larger numbers and the longer time to restart DB2 when there is an abnormal termination. The default value of 50 000 is good to start with. However, the value is completely application dependent, and LOGLOAD must be adjusted to keep DB2 restarts in the range of 10 minutes.

Locking Activity

The locking activity block shows information about number of lock and unlock requests, drain and claim requests, and also suspensions, timeouts, deadlocks, and lock escalations, if any, experienced in the reporting interval.

```
LOCKING ACTIVITY  QUANTITY
----------------  --------
DEADLOCKS             0.00
TIMEOUTS              0.00
SUSPENSIONS-LOCK      0.00
SUSPENSIONS-OTHR      0.00
LOCK REQUESTS      3013.00
UNLOCK REQUEST     1353.00
LOCK ESCALAT(SH)      0.00
LOCK ESCALAT(EX)      0.00
DRAIN REQUESTS       21.00
CLAIM REQUESTS      678.00
```

Figure 137. Statistics—Locking Activity

DEADLOCKS

The number of times lock suspensions ultimately resulted in a deadlock. This happens when two or more application processes each hold locks on resources that the others need and without which they cannot proceed. Deadlocks result principally from an application design problem. Ensure that all applications accessing the same tables access them in the same order. Deadlocks can also occur through index page splits if there is high insert activity. In this case, it is suggested to set SUBPAGES to 1 for the index.

TIMEOUTS

The number of times lock suspensions ultimately resulted in a timeout. This happens when a requester for a lock on a resource has waited longer than the installation-specified RESOURCE TIMEOUT limit on panel DSNTIPI.

SUSPENSIONS-LOCK

The number of resource conflicts. A suspension is a wait for a lock and each of these waits might contribute adversely to DB2 performance. The suspension can ultimately result in normal resumption, or, in the case of lock contention, in a timeout or deadlock. The number of lock suspensions is a function of the lock requests. Lock suspensions (or conflicts) can occur on either lock request or change request.

LOCK REQUESTS

The number of times a lock on a resource was requested.

LOCK ESCALAT(SH)

Count of lock escalations to shared mode. Number of times the LOCKS PER TABLE(SPACE) parameter on the panel DSNTIPJ was exceeded and the table space lock was promoted from a page lock (IS) to a table space lock (S). Escalation can cause unpredictable response times. The lock escalation to shared mode should only happen on an exception basis. For example, a REPEATABLE READ application references most pages in a table.

LOCK ESCALAT(EX)

Count of lock escalations to exclusive mode. Number of times the LOCKS PER TABLE(SPACE) parameter on the panel DSNTIPJ was exceeded and the table space lock was promoted from a page lock (IX) to a table space lock (X). Escalation can cause unpredictable response times. The lock escalation to exclusive mode should only happen on an exception basis. For example, when an application updates most pages in a table.

A useful rule of thumb is to compare the number of escalations (shared and exclusive) to the successful escalations (those that did not cause deadlocks and timeouts). In a single thread environment, if LOCK ESCALATION(SH) or LOCK ESCALATION(EX) is not zero, and if TIMEOUTS or DEADLOCKS is also not zero, then perhaps it can be concluded that the timeout or deadlock is caused by the escalation. However, in a real multi-thread environment, TIMEOUTS or DEADLOCKS are rarely zero (regardless of lock escalation). If many escalations cause deadlocks and timeouts, the recommendation is to change the escalation threshold value. Use of ANY is extremely desirable to prevent unnecessary and expensive page locks, for example, locking all pages in a tablespace.

Lock escalations, shared or exclusive, should not be expected in a transaction environment.

LOCKING CONSIDERATIONS

The following aspects should be considered if concurrency is an issue:
- Application design
- IRLM startup procedure options and DB2 installation options
- DDL LOCKSIZE
- BIND parameters.

Refer to *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* for detailed information.

EDM Pool Activity

Use the EDM pool block to gauge the efficiency of I/O on the EDM pool in accessing and retaining the most frequently used plans, packages, and database descriptors.

This block is shown in its LONG layout, because it contains additional fields that might be of interest at some sites.

```
EDM POOL                     QUANTITY  /MINUTE  /THREAD  /COMMIT
---------------------------  --------  -------  -------  -------
PAGES IN EDM POOL             1596.00      N/A      N/A      N/A
% PAGES IN USE                  10.72
 % NON STEAL. PAGES IN USE       7.98
FREE PAGES IN FREE CHAIN      1424.92      N/A      N/A      N/A
PAGES USED FOR CT                7.49      N/A      N/A      N/A
PAGES USED FOR DBD              41.49      N/A      N/A      N/A
PAGES USED FOR SKCT              5.00      N/A      N/A      N/A
PAGES USED FOR PT               44.32      N/A      N/A      N/A
PAGES USED FOR SKPT             72.78      N/A      N/A      N/A

FAILS DUE TO POOL FULL           0.00     0.00     0.00     0.00

REQUESTS FOR CT SECTIONS        21.00     0.36     1.00     0.11
CT NOT IN EDM POOL               5.00     0.09     0.24     0.03
CT REQUESTS/CT NOT IN EDM        4.20
REQUESTS FOR PT SECTIONS       709.00    12.09    33.76     3.83
PT NOT IN EDM POOL             159.00     2.71     7.57     0.86
PT REQUESTS/PT NOT IN EDM        4.46
REQUESTS FOR DBD SECTIONS      175.00     2.98     8.33     0.95
DBD NOT IN EDM POOL              6.00     0.10     0.29     0.03
DBD REQUESTS/DBD NOT IN EDM     29.17

PAGES IN DATASPACE               0.00      N/A      N/A      N/A
FREE PAGES IN DATASPACE          0.00      N/A      N/A      N/A
FAILS DUE TO DATASPACE FULL      0.00     0.00     0.00     0.00

PREP_STMT_CACHE_INSERTS          0.00     0.00     0.00     0.00
PREP_STMT_CACHE_REQUESTS         0.00     0.00     0.00     0.00
PREP_STMT_CACHE_PAGES_USED       0.00     0.00     0.00     0.00
PREP_STMT_HIT_RATIO               N/C
```

Figure 138. Statistics—EDM Pool Activity

% PAGES IN USE
The percentage of the EDM pool pages that are in use. If this percentage is consistently less than 50%, the EDM pool size is probably too large. The size can be reduced without affecting the efficiency ratios significantly. However, driving the EDM pool towards 100% usage can cause performance problems.

Note that EDM pool use will vary across the day. It is not only the pages that are in use, but also the pages that are not in use, waiting to avoid I/O, that are important for performance.

CT REQUESTS/CT NOT IN EDM
Ratio of number of requests for CT sections and number of times CT sections were not already in the EDM pool.

PT REQUESTS/PT NOT IN EDM
Ratio of number of requests for PT sections and number of times PT sections were not already in the EDM pool.

DBD REQUESTS/DBD NOT IN EDM
Ratio of number of requests for DBDs and number of times DBDs were not already in the EDM pool.

Online transactions cannot afford unnecessary I/O, which is the critical performance factor for transactions. Therefore, the necessary SKCTs, SKPTs, and DBDs should already be in the EDM pool whenever they are needed for a new thread. This means that the EDM pool must be large enough to prevent these from being stolen. The important values to monitor are those involved in the simple ratios discussed above, indicating EDM pool efficiency.

Rules of thumb:

- The above ratios could approach a high value with high volume, rereferencing transactions.
- For other environments a value of 5 to 10 for these ratios is acceptable.

PAGES IN DATASPACE
The number of pages in the data space used by the EDM pool.

FREE PAGES IN DATASPACE
The number of free pages in the free chain of the EDM pool data space.

FAILS DUE TO DATASPACE FULL
The number of failures because the EDM pool data space is full.

Log Activity

Use the log activity block to gauge the efficiency of DB2 logging activity. The information can be used to tune the size of active logs, size of log output buffer, and the write threshold value.

```
LOG ACTIVITY                       QUANTITY
---------------------------------  --------
READS SATISFIED-OUTPUT BUFFER        110.00
READS SATISFIED-ACTIVE LOG             0.00
READS SATISFIED-ARCHIVE LOG            0.00

READ DELAYED-UNAVAILABLE RESOURCE      0.00
ARCHIVE LOG READ ALLOCATION            0.00
ARCHIVE LOG WRITE ALLOCAT.             0.00
BSDS ACCESS REQUESTS                  56.00

WRITE OUTPUT LOG BUFFERS           25520.00
UNAVAILABLE OUTPUT LOG BUFFER          0.00
OUTPUT LOG BUFFER PAGED IN             0.00

LOG RECORDS CREATED                  420.3K
LOG DATA CREATED (MB)                 75.10
LOG WRITE I/O REQUESTS                 0.00
LOG CI WRITTEN                         0.00
LOG CI SERIAL WRITE                    0.00
```

Figure 139. Statistics—Log Activity

READS SATISFIED-ARCHIVE LOG
The number of times DB2 needed to read log records, and had to go to the archive log for the records. The value for this should ideally be zero. For optimal performance, when the data is backed out, it should still be available in the output buffer or in the active log. If the data has already been offloaded to the archive log, the active log is probably too small.

WRITE OUTPUT LOG BUFFERS
The number of calls to the log write routine. This does not represent the number of physical log I/Os. This should have a value consistent with the known workload update rate.

UNAVAILABLE OUTPUT LOG BUFF
This field shows how many times a write request to the active log had to wait because no buffer was available. The value should ideally be zero as these waits should not occur. If these waits do occur, the output buffer might be too small, or the size of the write threshold might be too close to the size of the output buffer.

LOG DATA CREATED (MB)
The amount of log data created (expressed in megabytes). Log records are placed sequentially in output log buffers, which are formatted as VSAM control intervals. The control intervals are written to a set of predefined DASD active log data sets, which are used sequentially and recycled.

A useful ratio is: WRITE OUTPUT LOG BUFFER divided by CONTROL INTERVAL CREATED-ACTIVE.

Rules of thumb:

The lower the value, the better. A high value indicates that too many I/Os are required for the number of log buffers created.

It is possible that write threshold is set too low. It is also possible that transactions could be arriving so infrequently that at commit time force requests are not queued and each force request is individually triggering an I/O of its log buffers.

CONSIDERATIONS FOR LOGGING

Minimize device contention on the log data sets by placing data sets correctly, that is, if using dual logging, do not place both logs on the same volume.

Avoid waits that occur because no log buffer is available.

Define enough active log data sets to prevent DB2 from waiting while a log is archived.

Make the active logs large enough that backouts do not have to use the archive log.

Consider the 3990 DASD FAST WRITE controller for the log. Performance measurements have shown that sequential access mode with DASD FAST WRITE provided substantially better performance than native DASD when the amount of log data written per commit was 24 KB or less. DASD FAST WRITE performance was comparable to that of native DASD when 48 KB of log data was written to DASD for each commit. When more than 48 KB was written, native DASD performed better than DASD FAST WRITE. Therefore there could be a need to determine in which environments log performance is critical in order to assess the value of DASD FAST WRITE.

Buffer Pool Activity

Use the buffer pool block to gauge the efficiency of I/O on the buffer pool in database read and write operations, as well as work file activity.

The following example shows the buffer pool block on the short statistics report.

```
BP0    GENERAL        QUANTITY
--------------------  --------
BPOOL HIT RATIO (%)      25.52
HPOOL HIT RATIO (%)      10.00
HPOOL R/W RATIO (%)       9.00
GETPAGES-SEQ&RANDOM    3003.00
GETPAGES-SEQ.ONLY        25.00
SYNC.READ-SEQ&RANDOM    525.00
SYNC.READ-SEQ.ONLY       12.00
SEQ.PREFETCH REQ          8.00
SEQ.PREFETCH READS        7.00
PAGES READ-SEQ.PREF.     58.00
LST.PREFETCH REQUEST      0.00
LST.PREFETCH READS        0.00
PAGES READ-LST.PREF.      0.00
DYN.PREFETCH REQUEST      0.00
DYN.PREFETCH READS        0.00
PAGES READ-DYN.PREF.      0.00
BUFFER UPDATES          929.00
SYNCHRONOUS WRITES       20.00
ASYNCHRONOUS WRITES       9.00
DATA SET OPENS           28.00
HDW THRESHOLD             0.00
VDW THRESHOLD             0.00
DM THRESHOLD              0.00
```

Figure 140. Statistics—Buffer Pool Activity

BPOOL HIT RATIO (%)
The total number of GETPAGE operations, minus the number of pages read from DASD (both synchronously and using prefetch), divided by the total number of GETPAGE operations, multiplied by 100.

HPOOL HIT RATIO (%)
This field is a means of measure for I/O avoidance due to the hiperpool. It is the percentage of pages moved into the virtual pool that could be retrieved from the hiperpool instead of reading them from DASD. The field is calculated as follows:

(Pages read from hiperpool / (Pages read from hiperpool + Pages read from DASD)) * 100

HPOOL R/W RATIO (%)
This field is a means of measure for how efficiently pages are retrieved from the hiperpool. The field is calculated as follows:

The number of pages read from hiperpool divided by the number of pages written to hiperpool multiplied by 100.

GETPAGES-SEQ&RANDOM
The number of GETPAGEs issued by sequential requests.

SYNC.READ-SEQ&RANDOM
The number of synchronous read I/O operations performed by sequential requests.

SEQ.PREFETCH READS
The number of times sequential prefetch reads were done.

PAGES READ-SEQ.PREF.
The number of pages read due to sequential prefetch.

LST.PREFETCH READS
The number of times list prefetch reads were done.

PAGES READ-LST.PREF.
The number of pages read due to list prefetch.

DYN.PREFETCH READS
The number of times dynamic prefetch reads were done.

PAGES READ-DYN.PREF.
The number of pages read due to dynamic prefetch performed because of sequential detection.

SYNCHRONOUS WRITES
Total number of immediate writes for a page.

An immediate write can occur when:
- Any synchronous write is triggered
- An immediate write threshold is reached
- No deferred write engines are available.

HDW THRESHOLD
The number of times the deferred write threshold was reached. This threshold is a percentage of the virtual buffer pool that might be occupied by unavailable pages, including both updated pages and pages in use.

When this threshold is reached, the data sets with the oldest updated pages are written asynchronously until the number of unavailable buffers reaches 10% below the threshold.

VDW THRESHOLD
The number of times the vertical deferred write threshold was reached. This threshold is expressed as a percentage of the virtual buffer pool that might be occupied by updated pages from a single data set.

When this threshold is reached, writes are scheduled for that data set.

DM THRESHOLD
The number of times the data manager threshold was reached. This fixed threshold is experienced if 95% of the pages in the buffer pool are unavailable. This has a significant effect on CPU usage.

Miscellaneous Buffer Pool Fields

There are various other fields that are not part of the default short report, but can be included by using user-tailored reporting. For more information, refer to “Tailoring Report Layouts” on page 115.

Buffer Pool Management

Through DSNZPARM, the user specifies the virtual buffer pool (and hiperpool) sizes for the buffer pools BP0 through BP49 and BP32K through BP32K9. These values are shown in DB2 PM system parameters report. The buffer pool sizes can be changed dynamically by executing an ALTER BUFFERPOOL command specifying buffer pool attributes. DB2 PM system parameters report shows the desired size of each buffer pool and the statistics report provides information about the actual number of buffers allocated.

The statistics reports show if any thresholds were experienced. However, the system parameters report has to be generated if the user wants to know the threshold values. This is because the threshold values for horizontal deferred write threshold, vertical deferred write threshold, virtual buffer pool sequential steal threshold, hiperpool sequential steal threshold, and virtual buffer pool parallel sequential steal threshold can be set or altered by executing ALTER BUFFERPOOL command. The threshold values for sequential prefetch threshold (90%), data manager threshold (95%), and immediate write threshold (97.5%) are fixed.

Buffer Pool Considerations

- Use the DISPLAY BUFFERPOOL command for online monitoring.
- Use the ALTER BUFFERPOOL command for dynamic tuning.

	Default	**Range**	**If Zero**
Horizontal deferred write threshold	50% of virtual buffer pool	0 to 90%	32 × 4 KB or 4 × 32 KB page threshold
Vertical deferred write threshold	10% of virtual buffer pool	0 to horizontal deferred write threshold	32 × 4 KB or 4 × 32 KB page threshold
Virtual buffer pool sequential steal threshold	80% of virtual buffer pool	0 to 100%	Disables prefetch
Virtual buffer pool parallel sequential steal threshold	50% of virtual buffer pool sequential steal threshold	0 to 100%	Disables CP and Sysplex parallel processing and I/O parallelism)
Hiperpool sequential steal threshold	80% of hiperpool	0 to 100%	Hiperpool not used for sequential processing
Virtual buffer pool size	BP0=2000	56 to 400000 0 to 400000	BP0 Other 4 KB BP
Hiperpool size	No hiperpool	0 to 2 million buffers 0 to 250000 buffers	If 4 KB page If 32 KB page
CASTOUT	YES		

Example of a Multiple Buffer Pool Assignment

This is an example for dynamic performance monitoring and tuning as well as good performance if there are more than 10 000 buffers available.

- BP0: Catalog and directory
  - Defaults generally are OK.
- BP1: Work files
  - Dedicated buffer pool ideal for work files if large sort or many concurrent sorts.
  - Try to allocate sufficient buffers if sort performance is important. Watch out for the following buffer pool fields:
    - MERGE PASS DEGRADED-LOW BUF
    - WORKFILE REQ REJCTD-LOW BUF
    - WORKFILE NOT CREATED-NO BUF
    - WORKFILE PRF NOT SCHEDULED.
  - Virtual buffer pool sequential steal threshold = hiperpool sequential steal threshold = 100.
- BP2: Frequently accessed tables that are relatively small with not much update
  - Allocate sufficient virtual buffer pool and hiperpool buffers to eliminate read I/O.
- BP3: Frequently updated tables that are relatively small
  - Allocate sufficient virtual buffer pool buffers to eliminate read and write I/O.
  - No hiperpool.
  - Vertical deferred write threshold = horizontal deferred write threshold = 90.
- BP4: Other tables with small buffer re-reference
  - Allocate small buffers enough to enable prefetch.
  - No hiperpool.
  - If update, set vertical deferred write threshold and horizontal deferred write threshold to 0 to:
    1. Maintain continuous deferred writes to avoid the “hiccup” effect at checkpoint
    2. 32 × 4 KB or 4 × 32 KB pages written when accumulated
    3. Avoid write miss in DASD Fast Write.
- BP5: Indexes
  - Allocate all remaining buffers to this buffer pool.
  - Ideally, this buffer pool should have the largest number of buffers.

Hiperpool Considerations

The following hiperpool considerations are based on test measurements and values can be different in other environments.

- If the system on which DB2 is installed has the Asynchronous Data Mover Facility of MVS installed, you have the option of using hiperspaces to extend DB2's virtual buffer pools.
- Benefits of hiperpool:
  - Reduces read I/O by exploiting expanded storage beyond 2 GB.
  - Controls the size of central storage without increasing CPU time.
- There is no caching into hiperpool in the following cases:
  - Work file merge
  - Load, reorg, recover
  - Prefetch in query CP and Sysplex parallel processing and in I/O parallelism (except sort/merge)
  - Sequential read when hiperpool sequential steal threshold is 0.
- Hiperpool performance considerations comparing {virtual buffer pool=2X} versus {virtual buffer pool=X and hiperpool=X} in the following six cases:
  1. Central storage < 2X and hit in central storage: No difference.
  2. Central storage < 2X and hit in expanded storage: For virtual buffer pool = 2X, SMF CPU time is less, but the total RMF CPU time can be higher.
  3. Central storage < 2X and I/O: For virtual buffer pool=2X, more CPU time; hiperpool with much larger expanded storage can further reduce I/Os.
  4. Central storage => 2X and hit in central storage: No difference.
  5. Central storage => 2X and hit in expanded storage: Hiperpool takes up more CPU time, but not all of central storage is used.
  6. Central storage => 2X and I/O: Hiperpool takes up more CPU time (less compared to e. above) but not all of central storage used; adding more expanded storage could help reduce I/Os, with a possibly better price/performance.

  In summary:
  - For the same amount of central storage, hiperpool offers equal or better performance.
  - Hiperpool offers potential for:
    - Reduced I/O with bigger expanded storage
    - Improved price/performance.

Hiperpool Considerations - continued

- Performance tuning suggestions:
  – Fully back up virtual buffer pool with central storage.

    Do not assign virtual buffer pool size > available central storage, instead, use hiperpool and reduce virtual buffer pool size. For example:
    - 256 MB central storage and 1024 MB expanded storage system
    - 50% availability to DB2
    - 50% of 50% available to DB2 buffer pools
    - Then, without hiperpool, virtual buffer pool=320 MB (64 MB central storage + 256 MB expanded storage)
    - With hiperpool, virtual buffer pool=64 MB and hiperpool=256 MB
      - Equal or better performance (cases 1, 2, and 3)
      - Controlled use of central storage.
    - May need to adjust horizontal deferred write threshold.

  – Make the size of the hiperpool at least twice the size of virtual buffer pool. Hiperpool smaller than virtual buffer pool increases CPU time without benefit.

Group Buffer Pool Activity

The Group Buffer Pool Activity blocks are only printed if the specific group buffer pool is connected to the reported DB2 system. Note that the counters are cumulative from the time the buffer pool was connected for the first time. If more than one 4 KB group buffer pool block is printed, then a block showing the 4 KB group buffer pool totals is printed. If more than one 32 KB group buffer pool block is printed, then a block showing the 32 KB group buffer pool totals is printed. If the report contains both 4 KB and 32 KB group buffer pool blocks, then a block showing the totals of all these blocks is printed.

An example of the group buffer pool activity block is shown in Figure 141.

```
GROUP BP0                    QUANTITY
---------------------------  --------
GROUP BP HIT RATIO (%)          50.00
SYN.READ(XI)-DATA RETURNED       0.00
SYN.READ(XI)-NO DATA RETURN      0.00
SYN.READ(NF)-DATA RETURNED       0.00
SYN.READ(NF)-NO DATA RETURN      0.00
CLEAN PAGES SYN.WRTN             0.00
CHANGED PGS SYN.WRTN            12.00
CLEAN PAGES ASYN.WRT             0.00
CHANGED PGS ASYN.WRT             0.00
REG.PG LIST (RPL) RQ            86.00
CLEAN PGS READ RPL               0.00
CHANGED PGS READ RPL             0.00
PGS READ FRM DASD AFTER RPL      0.00
ASYN.READ-DATA RETURNED          0.00
PAGES CASTOUT                    7.00
EXPLICIT X-INVALIDATIONS         0.00
CASTOUT CLASS THRESH             0.00
GROUP BP CAST.THRESH             0.00
CASTOUT ENG.UNAVAIL.             0.00
WRITE ENG.UNAVAIL.               0.00
READ FAILED-NO STOR.             0.00
WRITE FAILED-NO STOR             0.00
```

Figure 141. Statistics—Group Buffer Pool Activity

The following fields are contained in the group buffer pool activity block:

GROUP BP HIT RATIO (%)
This field reflects the percentage of pages that have been retrieved successfully from the Group Buffer Pool. The field is calculated as follows:

(The sum of pages read from the GBP / (the sum of pages read from the GBP + the sum of pages read from DASD)) * 100

SYN.READ(XI)-DATA RETURNED
The number of synchronous coupling facility read requests caused by the page in the member's buffer pool that is marked *invalid*. Data is returned from the group buffer pool.

SYN.READ(XI)-NO DATA RETURN
The number of synchronous coupling facility read requests due to the buffer being marked *invalid*. Data is not returned from the group buffer pool.

SYN.READ(NF)-DATA RETURNED
The number of synchronous coupling facility read requests necessary because the requested page was not found in the buffer pool. Data was returned from the coupling facility.

SYN.READ(NF)-NO DATA RETURN
The number of synchronous coupling facility read requests necessary because the requested page was not found in the buffer pool. Data is not returned from the coupling facility.

CLEAN PAGES SYN.WRTN
The number of clean pages synchronously written from the member's virtual pool to the group buffer pool.

CHANGED PGS SYN.WRTN
The number of changed pages synchronously written from the member's virtual pool to the group buffer pool.

CLEAN PAGES ASYN.WRT
The number of clean pages asynchronously written from the member's virtual pool to the group buffer pool. If a buffer pool threshold is reached, pages can be forced out before the application commits. This can also happen when P-lock negotiation forces the pages on the vertical deferred write queue to be written to the group buffer pool.

CHANGED PGS ASYN.WRT
The number of changed pages asynchronously written from the member's virtual pool to the group buffer pool.

REG.PG LIST (RPL) RQ
The number of requests to register a page list in the coupling facility.

CLEAN PGS READ RPL
The number of coupling facility reads to retrieve a clean page from the group buffer pool.

CHANGED PGS READ RPL
The number of coupling facility reads to retrieve a changed page from the group buffer pool.

PGS READ FRM DASD AFTER RPL
The number of pages that were read from DASD after an RPL request.

ASYN.READ-DATA RETURNED
The number of coupling facility reads for prefetch, in which data was returned from the coupling facility.

PAGES CASTOUT
The number of pages castout from the group buffer pool to DASD.

EXPLICIT X-INVALIDATIONS
The number of times an explicit coupling facility cross-invalidation request was issued.

CASTOUT CLASS THRESH
The number of times group buffer pool castout was initiated due to the class castout threshold being detected.

GROUP BP CAST.THRESH
The number of times group buffer pool castout was initiated due to the group buffer pool castout threshold being detected.

CASTOUT ENG.UNAVAIL.
The number of times the castout engine was not available.

WRITE ENG.UNAVAIL.
The number of times that a coupling facility write engine was not available for coupling facility writes.

READ FAILED-NO STOR.
The number of coupling facility read requests that could not complete due to a lack of coupling facility storage resources. If the value of this counter is constantly high, consider increasing the group buffer pool size.

WRITE FAILED-NO STOR
The number of coupling facility write requests that could not complete due to a lack of coupling facility storage resources. If the value of this counter is constantly high, consider increasing the group buffer pool size.

RID List Processing

The RID list processing block shows if the RID list storage size is adequate or should be increased.

```
RID LIST              QUANTITY
--------------------  --------
MAX BLOCKS ALLOCATED      0.00
CURRENT BLKS ALLOC.       0.00
FAILED-NO STORAGE         0.00
FAILED-RDS LIMIT          0.00
FAILED-DM LIMIT           0.00
FAILED-PROCESS LIMIT      0.00
```

Figure 142. Statistics—RID List Processing

FAILED-NO STORAGE
The number of times RID list processing was terminated due to insufficient storage.

This failure occurs when 2 GB limit is reached. You cannot increase virtual storage beyond 2 GB.

FAILED-RDS LIMIT
The number of times RID list processing was terminated because the number of RID entries was greater than the maximum limit of 25% of the table size or because the number of RID entries that can fit into the guaranteed number of RID blocks was exceeded. The latter can only happen when MAX BLK ALLOC approaches the RID list storage size. There is one guaranteed RID block.

FAILED-DM LIMIT
The number of times RID list processing was terminated because the number of RID entries exceeded the Data Manager limit of 16 million.

FAILED-PROCESS LIMIT
The number of times RID list processing was terminated due to the maximum RID list storage used.

The size is determined by the installation parameter RID POOL SIZE (DB2 install panel DSNTIPC). It can be 0, or between 16 KB and 1 GB.

The general formula for calculating the RID pool size is:

```
Number of concurrent RID processing activities ×
average number of RIDs × 2 × 5 bytes per RID
```

Open/Close Activity

The open/close activity processing block shows the efficiency in the handling of data sets and, if it warrants changing, the CLOSE rule for the page sets.

OPEN/CLOSE ACTIVITY	QUANTITY
OPEN DATASETS - HWM	83.00
OPEN DATASETS	68.00
IN USE DATA SETS	3.00

Figure 143. Statistics—Open/Close Activity

OPEN DATASETS - HWM

The maximum number of data sets open concurrently (high water mark).

Monitor this field to see whether you are reaching the maximum number of open data sets permissible. The maximum number of open data sets depends on the MVS version, modifications to your system, and other various factors. Refer to message IEF773I issued at IPL time for the maximum number of open data sets permissible for your system.

DDF Activity

The DDF activity processing block shows if block fetch is being effectively used, and any VTAM tuning is warranted.

REMOTE LOCATION	TRN SENT TRN RECV	CON SENT CON RECV	CON QUE SYS BIND	SQL SENT SQL RECV	COM SENT COM RECV	RBK SENT RBK RECV	ROW SENT ROW RECV	MSG SENT MSG RECV	BYT SENT BYT RECV	LIM BLK MSG BUF	BLK SENT BLK RECV
	78	78	78	3141	522	0	0	4302	537870	0	0
DRDA REMOTE LOCS	0	0	0	0	0	0	538493	3700	13274221	538369	711
	2	2	2	53	5	0	79	289	112933	0	0
J93DB22	0	0	23	192	0	0	9	327	124184	0	0

Figure 144. Statistics—DDF Activity

REMOTE LOCATION

Location name of the remote location with which DDF data is associated, or, for DRDA protocol, this field contains the string ‘DRDA REMOTE LOCS’. Statistics for all remote locations accessed by DRDA protocol are grouped under the location name ‘DRDA REMOTE LOCS’. For DB2 private protocols, statistics are gathered independently for each remote location.

CON QUE

The number of conversation requests, queued by the Distributed Data Facility, waiting for allocation.

When this value is high, you may want to increase the number of conversations by tuning VTAM.

LIM BLK

The number of times a switch was made from continuous to limited block fetch mode. This value applies only to DB2 private protocol.

When this value is high, you may want to increase the number of conversations by tuning VTAM.

Data Sharing Locks

```
DATA SHARING LOCKS  QUANTITY
------------------  --------
GLB CONT.RATE (%)       1.25
FLS CONT.RATE (%)      30.25
L-LOCKS RATE (%)        0.25
LOCK REQ.(P-LOCK)     296.00
UNLOCK REQ.(P-LCK)     58.00
CHANGE REQ.(P-LCK)    191.00
SYNC.XES - LOCK      2391.00
SYNC.XES - CHANGE     326.00
SYNC.XES - UNLOCK    2087.00
ASYN.XES-RESOURCES      0.00
TOTAL SUSPENDS          0.00
P-LCK/NFY ENG.UNAV      0.00
INCOM.RETAINED LCK      0.00
PSET/PART NEGOTIAT      0.00
PAGE NEGOTIATION        0.00
```

Figure 145. Statistics—Data Sharing Locks

The following fields are contained in the data sharing locking block:

GLB CONT.RATE (%)
The total number of suspends because of contention, divided by the total number of requests that went to XES (excluding asynchronous requests), multiplied by 100.

Aim for a total global lock contention of less than 5%, preferably less than 2%.

FLS CONT.RATE (%)
The number of false contentions, divided by the total number of contentions, multiplied by 100. A false contention is where two different locks on different resources hash to the same lock entry.

Try to keep the false contention rate below 50% of the total global lock contention.

L-LOCKS RATE (%)
The number of Transaction Locks (L-Locks) that have been propagated to XES and the Coupling Facility, expressed as a percentage of lock requests. This value does not include requests that are suspended.

LOCK REQ.(P-LOCK)
The number of lock requests for physical locks.

UNLOCK REQ.(P-LCK)
The number of unlock requests for physical locks.

CHANGE REQ.(P-LCK)
The number of change requests for physical locks.

SYNC.XES - LOCK
The number of logical and physical lock requests propagated to MVS XES synchronously.

SYNC.XES - CHANGE
The number of logical and physical change requests propagated to MVS XES synchronously.

SYNC.XES - UNLOCK
The number of logical and physical unlock requests propagated to MVS XES synchronously.

ASYN.XES-RESOURCES
The number of resources propagated by IRLM to MVS XES asynchronously. This number indicates the number of lock-related requests that were propagated to XES asynchronously. In DB2, the term *asynchronous* means that the request was handled under a system execution unit that runs asynchronously to the allied work unit.

This particular counter can be incremented, for example, when one DB2 subsystem has an IS lock on a particular table space, and another DB2 subsystem requests an IX lock. The S child locks held by the first DB2 must be propagated under a system execution unit to XES and the coupling facility.

TOTAL SUSPENDS
The total number of suspensions caused by IRLM global resource contention, MVS XES global resource contention, and false contention.

P-LCK/NFY ENG.UNAV
The number of times an engine is not available for physical lock exit or notify exit requests.

INCOM.RETAINED LCK
The number of global lock or change requests denied due to an incompatible retained lock.

PSET/PART NEGOTIAT
The number of times this DB2 was driven to negotiate a partition/pageset physical lock due to changing inter-DB2 interest levels on the partition/pageset.

PAGE NEGOTIATION
The number of times this DB2 was driven to negotiate a page physical lock due to physical lock contention within DB2.

Stored Procedures

```
STORED PROCEDURES  QUANTITY
-----------------  --------
CALL STATEMENTS        0.00
PROCEDURE ABENDED      0.00
CALL TIMED OUT         0.00
CALL REJECTED          0.00
```

Figure 146. Statistics—Stored Procedures

CALL STATEMENTS
The number of SQL CALL statements executed.

PROCEDURE ABENDED
The number of times a stored procedure terminated abnormally.

CALL TIMED OUT
The number of times an SQL CALL statement timed out waiting to be scheduled.

CALL REJECTED
The number of times an SQL CALL statement was rejected due to the procedure being in the STOP ACTION(REJECT) state.

Query Parallelism

```
QUERY PARALLELISM      QUANTITY
--------------------   --------
MAX DEGREE                 0.00
GROUPS EXECUTED            0.00
 RAN AS PLANNED            0.00
 RAN REDUCED               0.00
 FALL TO SEQUENTIAL        0.00
 ONE DB2 COORD P=NO        0.00
 ONE DB2 ISO LVL           0.00
MEMBER SKIPPED (%)         0.00
REFORM PARAL-CONFIG        0.00
REFORM PARAL-NO BUF        0.00
```

Figure 147. Statistics—Query Parallelism

MAX DEGREE
Maximum degree of parallelism executed among all the parallel groups. This field indicates the extent to which queries were processed in parallel.

GROUPS EXECUTED
Total number of parallel groups that have been executed.

RAN AS PLANNED
Total number of parallel groups which executed to the planned parallel degree. This field is incremented by one for each parallel group that executed with the planned degree of parallelism (as determined by DB2).

RAN REDUCED
Total number of parallel groups that processed to a parallel degree less than planned because of a storage shortage or contention on the buffer pool. If this field is not zero, increase the size of the current buffer pool by using the ALTER BUFFERPOOL command, or use the ALTER TABLESPACE statement to assign table spaces accessed by this query to a different buffer pool.

FALL TO SEQUENTIAL
The total number of parallel groups that fall back to sequential mode.

ONE DB2 COORD P=NO
The total number of parallel groups executed on a single DB2 subsystem due to the COORDINATOR subsystem value being set to NO. When the statement was bound, the COORDINATOR subsystem value was set to YES. This situation can also occur when a package or plan is bound on a DB2 subsystem with COORDINATOR=YES, but is run on a DB2 subsystem with COORDINATOR=NO.

ONE DB2 ISO LVL
The total number of parallel groups executed on a single DB2 subsystem due to repeatable-read or read-stability isolation.

MEMBER SKIPPED (%)
The percentage of parallel groups that were not distributed over the data

sharing group because one or more DB2 members did not have enough buffer pool storage. This applies only to parallel groups that were intended to run in Sysplex query parallelism.

The purpose of this count is to indicate when there are insufficient buffers on a member. Therefore, this count is only incremented when the buffer pool is defined to allow parallelism. For example, if VPXPSEQT = 0 on an assistant, DB2 does not send parallel work there, and this count is not incremented.

REFORM PARAL-CONFIG
Total number of parallel groups in which DB2 reformulated the parallel portion of the access path due to one of the following:

- A change in the number of active DB2 members
- A change in the processor models on which they run from bind time to run time.

This counter is incremented only on the parallelism coordinator at run time.

REFORM PARAL-NO BUF
Total number of parallel groups in which DB2 reformulated the parallel portion of the access path due to insufficient buffer pool resources. This counter is incremented only on the parallelism coordinator at run time.

Triggers

```
TRIGGERS              QUANTITY
--------------------  --------
STATEMENT TRIGGER         0.00
ROW TRIGGER               0.00
SQL ERROR OCCURRED        0.00
```

Figure 148. Statistics—Triggers

STATEMENT TRIGGER
The number of times a statement trigger was activated.

ROW TRIGGER
The number of times a row trigger was activated.

SQL ERROR OCCURRED
The number of times an SQL error occurred during the execution of a triggered action. This includes errors that occur in user-defined functions or stored procedures that are called by triggers and pass back a negative SQLCODE.

ROW ID

```
ROW ID                QUANTITY
--------------------  --------
DIRECT ACCESS             0.00
INDEX USED                0.00
TS SCAN USED              0.00
```

Figure 149. Statistics—ROW ID

DIRECT ACCESS
The number of times direct row access was successful.

INDEX USED
The number of times an attempt to use direct row access reverted to using an index to locate a record.

TS SCAN USED
The number of times an attempt to use direct row access reverted to using a table space scan to locate a record.

Chapter 19. Monitoring SQL Activity

Tuning DB2

This chapter identifies and describes the specific DB2 data which is reported for the purpose of tuning DB2. For general tuning advice on DB2, refer to the DB2 Administration Guide ‘Performance, Monitoring, and Tuning’ chapters for the specific release of DB2.

The Sysplex query parallelism function allows parallel processing for both CPU and I/O-intensive read-only queries within a data sharing group. This means that DB2 can split a single read-only query and process it on several DB2 subsystems.

As a consequence, the activity for OPEN and SELECT statements that exploit Sysplex query parallelism can now be reported from several members. The DB2 PM SQL activity report set supports Sysplex query parallelism in the same manner as it supported CP query parallelism in earlier versions. The data from all parallel tasks, regardless on which data sharing member they were executed, is combined with the data of the originating task.

The following general rules apply:

- All threads that are contained in the input data sets are reported. However, the INCLUDE/EXCLUDE and FROM/TO options can be used to reduce the amount of data shown in a trace or report.
- For each qualifying thread in the input data sets that uses Sysplex query parallelism, all parallel activity is taken into account as long as the corresponding parallel records are included in the input.
- Parallel records in the input that do not have a qualifying originating record in the output are not taken into account.

The TCB times of the originating task and the parallel tasks might not be comparable if the tasks run on processors of different speed. Therefore, DB2 PM uses a conversion factor to adjust the TCB times of the parallel tasks, so that they are normalized to the processor speed of the originating task.

For queries exploiting Sysplex query parallelism, the TCB time per event shows only the TCB time of the originating task. The TCB times of the parallel tasks are included in the Query Parallelism Workload block. This is the block where the TCB times of the parallel tasks are normalized, so they can be compared with the TCB time of the originating task. Therefore, you can view the TCB times of the parallel tasks as “artificial” times.

This chapter shows how to interpret some of the key fields in the SQL Activity traces and reports. The fields are presented in the order they appear in the different SQL Activity workload blocks in the SQL Activity traces.

Use SQL Activity traces and reports when the accounting information indicates a problem with SQL processing, or you are prototyping SQL applications.

As an illustration, if the class 2 TCB time, the time spent in DB2, shown on the accounting reports, is perceived to be high for this application, the SQL Activity reports can be used to determine the processing activity for each SQL statement.

SQL Activity

Access strategy for each dynamic SQL statement and also SQL statements bound at run time because of VALIDATE(RUN) can be determined from the SQL Activity Minibind workload block.

Look at the SQL Activity report to identify SQL statement numbers, cursors, or programs with high TCB time for selected plans and authorization IDs.

Look at the SQL Activity trace to identify SQL statement occurrences, statement numbers, cursors, or programs with high TCB time for an individual application.

Here is an example of an SQL Activity trace summarized by cursor name and sorted by average TCB time with workload highlights:

```
   LOCATION: DSNAPC3                            DB2 PERFORMANCE MONITOR (V6)                            PAGE: 1-1
      GROUP: GROUP_3                                 SQL ACTIVITY - TRACE                      REQUESTED FROM: NOT SPECIFIED
     MEMBER: MEMBER_2                                                                                      TO: NOT SPECIFIED
  SUBSYSTEM: APC3                                                                                 ACTUAL FROM: 04/08/99 03:28:52.13
DB2 VERSION: V6
                                  SUMMARIZED BY CURSOR, SORTED BY AVERAGE TCBTIME, WITH HILITE WORKLOAD

             PRIMAUTH: XXASP09           CONNECT : BATCH              CORRNAME: XXASP09F   CONNTYPE: TSO
             ORIGAUTH: XXASP09           PLANNAME: LOCCURHL           CORRNMBR: 'BLANK'    THRDTYPE: ALLIED
             ENDUSER : 1234567890123456  WSNAME  : 123456789012345678  TRANSACT: 12345678901234567890123456789012

TRACE # 1.1                 DB2 LUWID: APCNET.SYDAPC3.X'A44FECDC9ED4'                        ACE ADDRESS: X'02684398'

START TIME: 04/08/99 03:28:52.13  START ELAPSED:              0.087542    START REASON: CREATE THREAD
STOP TIME : 04/08/99 03:29:02.63  STOP ELAPSED :              0.009044    STOP REASON : TERMINATE THREAD

       EVENT            COUNT     TOT.ELAPS  TOTAL TCB                                DETAIL
                                  AET/EVENT  TCB/EVENT
-------------------- ----------- ----------- --------- ---------------------------------------------------------------------------

C2NOHOLD                        3    7.853205  0.029523 DBRM: LOCCURHL
                                     2.617735  0.009841 STMTTYPE           COUNT   AET/OCCUR TCB/OCCUR              COMMITS:     4
                                                        UPDATE                 1    0.230461  0.004741
                                                        OPEN                   3    0.001238  0.000251
                                                        FETCH                  3    1.001623  0.002264
                                                        DELETE                 1    4.614161  0.017236
      --- WORKLOAD HILITE ----------------------------------------------------------------------------------------------------
      SCANS   :     8  RECS/SORT:  3.00  I/O REQS:        1  SUSPENDS   :        2  EXITS    :        2  AMS               :        1
      ROWSPROC:     8  WORK/SORT:  2.00  AET/I/O : 1.374752  AET/SUSP   : 0.485483  AET/EXIT : 0.048234  AET/AMS           : 0.094745
      PAGESCAN:    47  PASS/SORT:  2.00  DATACAPT:      YES  RIDS UNUSED:        2  CHECKCON : REJECTED  DEGREE REDUCTION :        3
      LOB_PAGSCAN:    12345         LOB_UPD_PAGE:    12345

C1HOLD                          1    1.187397  0.009101 DBRM: LOCCURHL
                                                        STMTTYPE           COUNT   AET/OCCUR TCB/OCCUR              COMMITS:     4
                                                        UPDATE                 2    0.466585  0.002298
                                                        OPEN                   1    0.000208  0.000207
                                                        FETCH                  4    0.063392  0.000962
                                                        CLOSE                  1    0.000451  0.000451
      --- WORKLOAD HILITE ----------------------------------------------------------------------------------------------------
      SCANS   :     8  RECS/SORT:  3.00  I/O REQS:        1  SUSPENDS   :        2  EXITS    :        2  AMS               :        1
      ROWSPROC:     8  WORK/SORT:  2.00  AET/I/O : 1.374752  AET/SUSP   : 0.485483  AET/EXIT : 0.048234  AET/AMS           : 0.094745
      PAGESCAN:    47  PASS/SORT:  2.00  DATACAPT:      YES  RIDS UNUSED:        2  CHECKCON : REJECTED  DEGREE REDUCTION :        3
      LOB_PAGSCAN:    12345         LOB_UPD_PAGE:    12345

#   245                         1    1.137912  0.007616 INSERT                                       ISO N/A
                                                        DBRM: LOCCURHL
      --- WORKLOAD HILITE ----------------------------------------------------------------------------------------------------
      SCANS   :     8  RECS/SORT:  3.00  I/O REQS:        1  SUSPENDS   :        2  EXITS    :        2  AMS               :        1
      ROWSPROC:     8  WORK/SORT:  2.00  AET/I/O : 1.374752  AET/SUSP   : 0.485483  AET/EXIT : 0.048234  AET/AMS           : 0.094745
      PAGESCAN:    47  PASS/SORT:  2.00  DATACAPT:      YES  RIDS UNUSED:        2  CHECKCON : REJECTED  DEGREE REDUCTION :        3
      LOB_PAGSCAN:    12345         LOB_UPD_PAGE:    12345
------------------------------------------------------------------------------------------------------------------------------------
```

Figure 150. SQL Activity Trace Summarized by Cursor, Sorted by Average TCB Time, with Hilite Workload

Average TCB time is shown for each statement occurrence executed under the cursor name as well as for the cursor as a whole.

For information on how to produce SQL Activity traces, refer to “SQL Activity” on page 82 and to the *DB2 PM Report Reference Volume 2*.

The following examples show various blocks of DB2 workload activity printed in SQL Activity traces. Note that, in a CP and Sysplex parallel processing situation, SQL Activity rolls the workload for any parallel tasks into that of the originating thread. The figures for Scan, Sort, I/O, RID List, Lock Suspension, Page and Row Locking, and Exit Activity thus reflect the work done by parallel tasks as well as by the originating task.

Scan Activity Workload Block

This block shows the total scan activity, per object, performed by the event.

```
--- SCAN ACTIVITY -------------------------------------------------------------------------------------------------------------
                                ------ROWS------  --QUALIFIED AT--  --------- -ROWS----------- --MASS- --PAGES- --------RI-------
DATABASE  PAGESET    SCANS   PROCESS  EXAMINE  STAGE 1  STAGE 2  INSERTS   UPDATES  DELETES DELETES  SCANNED    SCANS  DELETES
MEMBER    TYPE
DSNDB01   SCT02          6         6        6        6        0        0         0        0       0       12        0        0
SE12      INDX
DSNDB01   SCT02          6         6        6        0        0        0         0        0       0        4        0        0
SE12      SEQD
DSNDB06   SYSPLAN        1         1        1        1        0        0         0        0       0        2        0        0
SE12      INDX
DSNDB06   SYSPLAN        1         1        1        0        0        0         0        0       0        1        0        0
SE12      SEQD
NHDBASE1  NHTSPAC1       1         4        4        4        4        0         2        0       0        5        0        0
SE12      SEQD
TOTAL                   15        18       18       11        4        0         2        0       0       24        0        0
```

Figure 151. Example of an SQL Activity Scan Workload Block

ROWS PROCESS, ROWS EXAMINE and TYPE
: If the values in the ROWS PROCESS and ROWS EXAMINE fields do not match when TYPE is SEQD, check to see if a segmented table space can be used to reduce CPU and elapsed time.

QUALIFIED AT STAGE 1
: The total number of rows that were qualified at stage 1.

QUALIFIED AT STAGE 2
: The total number of rows that were qualified at stage 2. The value in this field cannot be greater than the value in QUALIFIED AT STAGE 1.

CPU cost is more if an SQL statement has stage-2 predicates to be evaluated. Check to see if the SQL statement can be written in such a way that only stage-1 predicates will be used.

Function Resolutions Information

The following is an example of the function resolutions information block:

```
--- FUNCTION RESOLUTION(S) -----------------------------------------------------------------------------------------------
QUERYNO  : 1383       PLANNAME : DSNTEP61  COLLECTION_ID : DSNTEP61
APPLNAME : xxxxxxxx   PROGNAME : DSNTEP61  CONSIS_TOKEN  : 1234567890123456
BIND_TIME: 04/08/98 03:28:55.21  VERSION : xxxxxxxx10xxxxxxxx20xxxxxxxx30xxxxxxxx40xxxxxxxx50xxxxxxxxxxxx64
CURRENT_PATH : xxxxxxxx10xxxxxxxx20xxxxxxxx30xxxxxxxx40xxxxxxxx50xxxxxxxx60xxxxxxxx70xxxxxxxx80xxxxxxxx90xxxxxxx100
               xxxxxxx110xxxxxxx120xxxxxxx130xxxxxxx140xxxxxxx150xxxxxxx160xxxxxxx170xxxxxxx180xxxxxxx190xxxxxxx200
               xxxxxxx210xxxxxxx220xxxxxxx230xxxxxxx240xxxxxxxxxxx254
....................................................................................................................
FUNCT_SCHEMA : SCHEMA01  FUNCT_NAME : xxxxxxxxxxxxxxxxxx  SPECIFIC_NAME : xxxxxxxxxxxxxxxxxx  FUNCT_TYPE : SCALAR
VIEW_CREATOR : NAME-111  VIEW_NAME  : xxxxxxxxxxxxxxxxxx  QUERY_BLOCKNO : 53
FUNCT_TEXT   : xxxxxxxx10xxxxxxxx20xxxxxxxx30xxxxxxxx40xxxxxxxx50xxxxxxxx60xxxxxxxx70xxxxxxxx80xxxxxxxx90xxxxxxx100
               xxxxxxx110xxxxxxx120xxxxxxx130xxxxxxx140xxxxxxx150xxxxxxx160xxxxxxx170xxxxxxx180xxxxxxx190xxxxxxx200
               xxxxxxx210xxxxxxx220xxxxxxx230xxxxxxx240xxxxxxxxxxx254
```

Figure 152. SQL Activity Function Resolutions Information Details

QUERYNO
The query number.

PLANNAME
The plan name.

COLLECTION_ID
The collection ID.

APPLNAME
The name of the application.

PROGNAME
The program name.

CONSIS_TOKEN
The consistency token.

BIND_TIME
The time stamp of the bind time.

VERSION
The version ID.

CURRENT_PATH
The current path. It can be up to 254 characters long.

The following function information part of the block is repeated for each function referenced in the statement:

FUNCT_SCHEMA
A short SQL identifier, either ordinary or delimited, following the concept of qualified names consistent with the ANSI/ISO SQL92 standard.

FUNCT_NAME
The name of a function without a qualifier.

SPECIFIC_NAME
Identifies the particular function. The specific name must identify a specific function name in the explicitly or implicitly specified schema.

FUNCT_TYPE
The classification of the function.

SCALAR
Scalar UDF

TABLE
Table UDF

VIEW_CREATOR
The name of the view creator if the function is referenced in a view definition.

VIEW_NAME
The name of the view if the function is referenced in a view definition.

QUERY_BLOCKNO
A number that identifies the query block number being explained.

FUNCT_TEXT
Contains the text of the function reference, function name, and parameters. It can be up to 254 characters long.

RID List Workload Block

This block shows RID list activity performed by the event.

Here is an example of the RID list processing workload block:

```
--- RID LIST PROCESSING -------------------------------------------------------------------------------------------------------
RIDS IN FINAL LIST:        38  RID LIST USED:     2  UNUSED (LIMIT EXEEDED):     5  UNUSED (NO STORAGE):     1
DATABASE  PAGESET    THRESHOLD RIDS OBTAINED        RIDS EXCEEDED LIMIT
NHDBASE1  NHINDEX1       4075            36                           3
NHDBASE2  NHINDEX2      36000            87                           2
           AVERAGE    20037.50         61.50                        2.50
```

Figure 153. SQL Activity RID List Workload Block

UNUSED (LIMIT EXCEEDED)
The number of times RID list processing was terminated due to limits exceeded. Find out from the trace what caused this condition and, if necessary, consider increasing the RID list storage size. Before doing this, check if the data is skewed as this might be the cause for too many RIDs being processed. Ensure RUNSTATS is current.

The size is determined by the installation parameter RID POOL SIZE (DB2 install panel DSNTIPC). It can be 0, or between 16 KB and 1 GB.

The general formula for calculating the RID pool size is:

```
Number of concurrent RID processing activities ×
average number of RIDs × 2 × 5 bytes per RID
```

UNUSED (NO STORAGE)
The number of times RID list processing was terminated due to insufficient storage. This failure occurs when 2 GB limit is reached.

Query Parallelism

This block shows query parallelism activity performed by the event.

Note: In query CP and Sysplex query parallelism, this is the only place where the TCB time of the parallel records is shown.

An example of a query parallelism workload block is shown in Figure 154.

SQL Activity

```
--- QUERY PARALLELISM ---------------------------------------------------------------------------
QUERY  PARALLEL  PLANNED  PLANNED  NEGOTIATED           PIPE            TASK            NUMBER OF
BLOCK  GROUP      AT BIND  AT  RUN  AT      RUN    REASON  ELAPSED TIME    CPU TIME    TYPE  MEMBERS
    1         1        3        3          3    NORMAL      0.895716    0.043467    CP            3
```

Figure 154. Query Parallelism Workload Block Example

QUERY BLOCK
The query block number.

PARALLEL GROUP
The parallel group number.

PLANNED AT BIND
The degree of parallelism planned at bind time.

This field contains 0 if host variables in the statement caused the parallelism decision to be made at bind time.

PLANNED AT RUN
The degree of parallelism planned at run time.

NEGOTIATED AT RUN
The degree of parallelism negotiated at run time, which depends on buffer pool availability.

If the value in this field is 1, the plan for parallel I/O processing falls back to sequential execution mode.

REASON
The reason for deriving the planned run-time degree of parallelism:

NORMAL
The planned run-time degree is derived from planned bind-time degree.

HOSTVAR
Host variable partitioning.

NOESA
No ESA sort support.

CURSOR
Cursor that can be used for update and delete.

EMPTY
Empty parallel group.

ENCLUNAV
MVS/ESA enclave services are not available.

UNKNOWN
None of the above.

PIPE ELAPSED TIME
The time of pipe creation subtracted from the time of pipe termination.

TASK CPU TIME
The sum of the normalized CPU times spent for the parallel tasks. In Sysplex query parallelism, the CPU times are normalized by the conversion factor that is derived from IFCID 106 and related to the conversion factor of the originating task.

If IFCID 106 is not present, asterisks are printed. For DB2 releases prior to DB2 for MVS/ESA Version 4, N/A is printed.

The task CPU time is calculated as follows:

- Let CV_O be the conversion factor for the member where the originating thread is running.
- Let CV_P be the conversion factor for the member where the parallel thread is running.
- Let TCB_P be the TCB time that is recorded by DB2 for an activity of the parallel thread.
- Then the following formula applies:

  Normalized TCB time for that activity = $(TCB_P * (CV_O / CV_P))$

TYPE The type of parallelism:

CP CP parallelism
I/O I/O parallelism
SYS Sysplex query parallelism

NUMBER OF MEMBERS
The number of members on which the query executed.

Sort Activity Workload Block

This block shows sort activity performed by SQL events.

Here is an example of the sort workload block:

```
--- SORT ACTIVITY ------------------------------------------------------------------------------------------
MEMBER      :      SE12  WORKFILES             :    72.00  RECORDS     :      2.45  MAX REQUESTED   :        5
TOTAL SORTS :         4  INITIAL WORKFILES     :     1.00  RECORD SIZE :  18292.50  AVG REQUESTED   :     3.35
SORT KEYS   :      2.00  WORKFILES PARTITIONED :     2.00  KEY SIZE    :     30.27  MAX NOT ACQUIRED:        1
SORT COLUMNS:     25.00  PARTITIONING          :      YES  DATA SIZE   :     92.13  AVG NOT ACQUIRED:     1.00
AET/SORT    :  2.990676  PARTITIONING & SORTING:      YES  ROWS DELETED:      0.00  MAX RETURN CODE :        4
SORT TYPE   :   ESA-TAG  PARTITION TYPE        : LASTPASS  MERGE PASSES:      1.00
```

Figure 155. SQL Activity Sort Workload Block

AET/SORT, RECORDS, KEY SIZE, RECORD SIZE, and WORKFILES
To minimize the sort time (AET/SORT), consider the following factors:

- Sort the minimum number of rows (RECORDS) by applying all possible predicates in the WHERE clause of the SELECT statement.
- Keep the sort key size as small as possible (KEY SIZE), especially avoiding sorting on VARCHAR columns as they are padded to their maximum length.
- Keep the row size to be sorted to a minimum (RECORD SIZE) by selecting only the columns of interest in the SELECT statement.
- Allocate as many sort work files across different DASD volumes as possible, unless constrained to a single volume when it is better to have a single large work file (WORKFILES).
- The number of merge passes for a given sort can be kept at no more than one by increasing sort workfile buffer pool size. So, if the number of merge passes is greater than 1, there is room for improvement.

I/O Activity Workload Block

This block shows the I/O activity, per object, performed by the event.

Here is an example of the I/O workload block:

```
--- I/O ACTIVITY -----------------------------------------------------------------------------------------------------------
DATABASE  PAGESET  - I/O REQUEST -  ------- READ REQUEST  (WITH OR WITHOUT I/O) -------  ------------ WRITE REQUEST ----------
MEMBER    BP       TOTAL     AET    TOTAL  TYPE   AET/WITH    %WITH PAGE/WITH  %WITHOUT  TOTAL  TYPE  CAST      AET  PAGE/WRIT
DBPARALL  TSPARALL
SE12      BP4          3  0.1296        3  SYNCH  0.129597   100.00      1.00     0.00
WRKSE12   DSN4K01
SE12      BP0        102  0.0164      102  SYNCH  0.016358   100.00      1.00     0.00
```

Figure 156. SQL Activity I/O Workload Block

I/O REQUEST AET
: Average elapsed time per I/O request for all types of read and write requests. A high average I/O elapsed time indicates I/O contention. DASD tuning is likely to be needed.

READ REQUEST PAGE/WITH
: Pages read per read request with I/O of a particular type. The maximum value for SQL activity is 32. A lower value might indicate a buffer pool shortage. If there are no buffer pool shortages, then a lower value might indicate that many of the desired pages are already in the buffer pool.

READ REQUEST %WITHOUT
: The percentage of total read requests without I/O for the particular type. A higher value (up to a maximum of 100) is good since it means that many of the pages requested by a prefetch read were already in the buffer pool.

WRITE REQUEST TOTAL, WRITE REQUEST TYPE, and WRITE REQUEST PAGE/WRITE
: The number of write I/O requests of a specific type, the type of write request, and the number of pages written per write of a specific type.

A value of SYNCH for TYPE should be rare and, when present, the value for WRITE REQUEST TOTAL for that entry should be low. If the value for WRITE REQUEST TOTAL is not low and the value for WRITE REQUEST PAGE/WRITE is much lower than 32, then probably buffer pool thresholds are being experienced or the frequency of DB2 checkpoints is high.

Lock Suspension Activity Workload Block

This block shows the lock suspension activity, per object, performed by the event.

Here is an example of the Lock Suspension workload block:

```
--- LOCK SUSPENSION ACTIVITY -----------------------------------------------------------------------------------------------
                                          ------------- SUSPEND --------------  NORMAL RESUME  TIMEOUT RESUME  DEADLOCK RESUME
RESOURCE NAME          TYPE      REQUEST  LOCAL LATCH IRLMQ GROUP NOTIFY OTHER  COUNT      AET  COUNT      AET  COUNT        AET
MEMBER
NHDBASE1     NHTSPAC1  DATAPAGE  CHANGE       1     0     0     0     0     0      0 13.8728      0      N/C      0        N/C
SE12
NHDBASE1     NHTSPAC2  DATAPAGE  LOCK         0     1     0     0     0     0      0 1.98182      0      N/C      0        N/C
SE12
NHDBASE1     XDEPT1    OPENLOCK  UNLOCK       1     0     0     0     0     0      0 4.98875      0      N/C      0        N/C
SE12
```

Figure 157. SQL Activity Lock Suspension Workload Block

This block can help identify the point in the application at which processing is being suspended. The resource, type of request, and the reason for the suspension can be seen in this block.

Page and Row Locking Workload Block

This block shows the page and row locking and lock avoidance activity, per object, performed by the event.

Here is an example of the page and row locking workload block:

```
--- PAGE & ROW LOCKING ------------------------------------------------------------------------
                                          LOCK   MAX LOCKS   # LOCK   HIGHEST    TS     LOCK AVOID
MEMBER    DATABASE   PAGESET   COUNT   SIZE              ESCAL      LOCK    TYPE   SUCCESSFUL
SE12      DSNDB06    SYSDBASE      2  TABLE          0        0        IS   SIMPL          N/P
SE12      DSNDB06    SYSDBAUT      2  TABLE          0        0        IS   PARTI          N/P
SE12      DSNDB06    SYSUSER       2    LOB          0        0         1     LOB          N/P
SE12      DSNDDF     SYSDDF        3  TABLE          0        0               SEG          YES
SUMMARY : MAX PAGE/ROW/LOB LOCKS HELD 12345   LOCK ESCALATIONS : SHARED 12345   EXCLUSIVE    12345

TOTAL                              9                        0
```

Figure 158. SQL Activity Page and Row Locking Workload Block

This block can help to identify if the locking problem experienced is triggered by the processing of this SQL statement. This block shows the page and row locking activity for each database page set accessed for the execution of the SQL statement. In detail, this block shows the following:

- The total number of page and row locks taken for the page set
- The maximum number of page or row locks held at any time for the page set
- Any escalations that have occurred from page-level or row-level locking as a result of reaching the value for either the LOCKS PER TABLE(SPACE) parameter of DSNZPARM or the LOCKMAX value for the tablespace listed
- The highest tablespace lock taken on the page set.

Exits Workload Block

This block shows the exits performed by the event.

Here is an example of the Exits workload block:

```
--- EXITS ---------------------------------------------------
MEMBER     VALIDATION  TOTAL   AET/EXIT   EDIT TOTAL  AET/EXIT
SE12                       1    N/C                0  0.000060
```

Figure 159. SQL Activity Exits Workload Block

Any validation or edit exits invoked can impact the timings in the SQL processing.

Data Capture Workload Block

This block shows average data capture activity performed by the event.

Here is an example of the Data Capture workload block.

```
--- DATA CAPTURE -------------------------------------------------------------
DESCRIBES     :      3.14  MAX READ TIME:  1.928397  DATA DESC RETURNED:     3.77
AET/DESCRIBE  :  0.028367  RECS RETURNED:     24.86  TABLES RETURNED   :     0.00
LOG READS     :      5.20  RECS CAPTURED:     29.15
AET/EXTRACTION:  1.044382  ROWS RETURNED:    132.50
```

Figure 160. SQL Activity Data Capture Workload Block

DESCRIBES
The average number of data capture describes.

AET/DESCRIBE
The average elapsed time of data capture describes.

LOG READS
The average number of log reads performed.

AET/EXTRACTION
The average elapsed time of log extraction.

MAX READ TIME
The longest elapsed time of a log read.

RECS RETURNED
The average number of log records returned.

RECS CAPTURED
The average number of records that were captured for this update. To perform all data capture updates, all captured log records need to be returned.

ROWS RETURNED
The average number of data rows returned.

DATA DESC RETURNED
The average number of data capture data descriptions returned.

TABLES RETURNED
The average number of data capture tables returned.

Minibind Workload Block

Product-Sensitive Programming Interface

This block shows the minibind activity performed by the event. Plans are generated by DB2 at BIND and SQL PREPARE time. One minibind entry is generated per subselect block in the query. If the query has subqueries, then more than one entry is written. Most fields in this record are also present in the plan table from an explain.

Here is an example of the minibind workload block with multiple index.

```
--- MINIBIND -------------------------------------------------------------------------------------------------------
QUERYNO  : 1383      PLANNAME      : DSNTEP61          COST          : 35        PARALLELISM_DISABLED : N/A
QBLOCKNO : 2         COLLID        : DSNTEP61          PROGNAME      : DSNTEP61  CONSISTENCY_TOKEN    : 15769AE806DB8B8E
APPLNAME : N/P       WHEN_OPTIMIZE : 'BLANK'           OPT_HINT_IDENT: N/P       OPTIMIZE_HINTS_USED  : YES
UNITS    : 12345     MILLI_SEC     : 12345             COST_CATEGORY : N/P
BIND_TIME: 04/08/98 03:28:55.211328  VERSION: N/P
....................................................................................................................
PLANNO          : 1                   METHOD    : FIRST TABLE ACCESSED  SORTN_UNIQ      : NO     SORTC_UNIQ     : NO
DATABASE        : DSNDB04             NEXTSTEP  : NOT APPLICABLE        SORTN_JOIN      : NO     SORTC_JOIN     : NO
OBJECT          : 21                  ACCESSTYPE: TABLE SPACE SCAN (R)  SORTN_ORDERBY   : NO     SORTC_ORDERBY  : NO
CREATOR         : X                   PAGE_RANGE      : NO              SORTN_GROUPBY   : NO     SORTC_GROUPBY  : NO
TNAME           : TBUF0401            JOIN_TYPE       : NO              SORTN_PGROUP_ID : 0      SORTC_PGROUP_ID: 0
CORRELATION_NAME: N/P                 MERGE_JOIN_COLS : 0               ACCESS_DEGREE   : 0      JOIN_DEGREE    : 0
TSLOCKMODE      : IS                  PARALLELISM_MODE: NO              ACCESS_PGROUP_ID: 0      JOIN_PGROUP_ID : 0
PREFETCH        : NO                  COLUMN_FN_EVAL  : N/P             PAGES_FOR_TABLE : 12345  TAB_CARDINALITY: 123456789A
DIRECT_ROW_ACC  : N/A
..................................................................................................................
   ACCESS_CREATOR   ACCESS_NAME          MATCHCOLS   INDEXONLY   PREFETCH_INDEX       OPERATION      MIXOPSEQ
   SYSADM           PXPS$PKSK            1           NO          NO                   SCAN           1
   SYSADM           UXS#SKNK             1           YES         NO                   SCAN           1
```

Figure 161. SQL Activity Minibind Workload Block with Multiple Index

If there is only one index, which is the case if field *MIXQPSEQ* is set to 1, then no separate table is listed on the bottom part of the Minibind workload block. Instead, the fields of this table are integrated in the repeating group of the block.

If there is no index at all, then the table fields are also integrated in the repeating group of the Minibind workload block, however, their values are set to N/A.

Refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide* and DB2 PM documentation on the explain function to understand the details of the access strategy as determined by DB2 and contained in the minibind record.

However, the following observations deserve attention:

- **METHOD**: FIRST TABLE ACCESSED, and **ACCESSTYPE**: TABLE SPACE SCAN, indicates index is not being used. Determine if this is satisfactory.
- **ACCESSTYPE**: INDEX SCAN, whatever **METHOD** might be, and **MATCHCOLS**: 0 indicates nonmatching index scan. Determine why it is not a matching index scan.
- **ACCESSTYPE**: INDEX SCAN, whatever **METHOD** might be, and **MATCHCOLS**: not 0 indicates matching index scan. Determine if it is a fully matching index or not.
- **METHOD**: NESTED LOOP JOIN and **ACCESSTYPE**: TABLE SPACE SCAN on the inner table can result in degradation of SQL Join performance, particularly if this table is large and the outer table’s qualifying rows are numerous. Determine why an index cannot be used.
- **METHOD**: MERGE SCAN JOIN and **SORTN_JOIN**: YES, make sure that sort itself is well tuned to facilitate the join.
- **METHOD**: HYBRID JOIN and **SORTN_JOIN**: YES indicates the inner table index is not clustered. You may want to consider reorganizing the inner table to avoid RID sort of the intermediate table. This can result in better join efficiency.

End of Product-Sensitive Programming Interface

Part 5. Appendixes

Appendix A. DB2 PM VSAM Data Sets

DB2 PM uses the following VSAM data sets:

- Save data sets are written when the job stream contains a SAVE subcommand.
- Job summary data sets are written when new data is processed.
- Distribute data sets are written when the job stream contains a DISTRIBUTE command.

All VSAM data sets used in a DB2 PM job must exist before DB2 PM is executed. Preallocate the data sets using the IDCAMS command. You can run IDCAMS as an initial step in the DB2 PM job. The required attributes for VSAM data sets are shown in Table 9. An example of the required IDCAMS commands is shown in Figure 162.

Refer to the *DFSMS/MVS Access Method Services for ICF* and the *DFSMS/MVS Access Method Services for VSAM* for more information about IDCAMS.

Notes:

1. When the SAVE subcommand is specified, the save data set should be empty. If it is not empty, all existing records are deleted. If save and restore use the same physical data set, the restored data is rewritten during save.
2. You need not prime DB2 PM VSAM data sets.

Table 9. Attributes for DB2 PM VSAM Data Sets

Data Set	Key Length (bytes)	Record Length (bytes)		Buffer Space (bytes)	Control Interval Size (bytes)	
		Maximum	Average		Data	Index
Accounting SAVE (ACSAVDD)	255	3000	1500	40 960	8192	4096
Statistics SAVE (STSAVDD)	92	5500	2000	40 960	8192	4096
DISTRIBUTE (DISTDD)	158	234	234	40 960	8192	4096
Job Summary (JSSRSDD)	52	2462	160	40 960	8192	4096

Note: Buffer space and control interval size are suggestions only. You can modify them to suit the requirements of your installation.

```
//ALCVSAM  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DELETE (cluster.name)
  DEFINE -
    CLUSTER ( -
      NAME (cluster.name) -
      TRACKS (as required) -
      VOLUMES (as required) -
      KEYS (keylength 0) -
      RECORDSIZE (average maximum) -
      BUFFERSPACE (40960) -
      REUSE -
      ) -
   DATA ( -
      CONTROLINTERVALSIZE (8192) -
      ) -
   INDEX ( -
      CONTROLINTERVALSIZE (4096) -
      )
/*
```

Figure 162. IDCAMS Commands

In this example, the job deletes the cluster if it already exists, then defines a new cluster with the specified attributes.

Appendix B. Using GROUP to Improve Save Data Set Performance

The standard key for DB2 PM save data sets contains a combination of the applicable DB2 PM identifiers from DB2 instrumentation records.

For example, the key for an accounting save record contains a number of DB2 PM 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 DB2 PM identifiers, you can use GROUP to reduce the uniqueness of the key, and thus the number of reduced data records on the save data set.

For example, if you only produce accounting reports using ORDER(CONNECT-PRIMAUTH-PLANNAME), you can use groups in INCLUDE REDUCE to get the minimum number of records in the save data set.

The following command stream does not use groups:

```
ACCOUNTING
     REDUCE
     SAVE
```

The save data set produced by this command stream contains reduced records for each unique combination of all of the DB2 PM identifiers in the key.

The following command stream uses groups to minimize the number of records in the save data set:

```
GROUP (CORRNAME (ALLCNM(*)))
GROUP (CORRNMBR (ALLCNU(*)))
GROUP (ORIGAUTH (ALLORI(*)))
GROUP (REQLOC   (ALLREQ(*)))
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 DB2 PM identifiers that are *not* grouped, for example, PRIMAUTH and PLANNAME.

During REDUCE processing, the group name is substituted for the original value on the key entry. For example, all records in the reduced data have a correlation name of ALLCNM. Using groups can significantly reduce the diversity of the keys in the save data, resulting in far fewer records, smaller save data sets, and improved performance.

Note, however, that because the key now contains the group name (not the original value), you cannot use the original values of any of the grouped DB2 PM identifiers in REPORT or INCLUDE/EXCLUDE in the same command stream or when you restore the data.

Appendix C. Comparing Original Authorization IDs with Primary Authorization IDs

This section 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(R) 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 at installation time
- The original (unaltered) value.

Notes:
1. The original authorization ID should be used when you are attempting to establish accountability of DB2 activity, since the primary authorization ID can be an ID other than the user (group name, for example).
2. If your subsystem uses authorization ID translation for distributed activity, the AUTHID reported for DBATs is the translated value.
3. For detailed information on authorization IDs, refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide*.

Comparing Secondary IDs with SQL ID

DB2 uses two other types of authorization IDs:
- Secondary authorization IDs
- SQL ID.

Authorization IDs

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 on authorization IDs, refer to the *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide*.

Appendix D. How to Use DB2 PM Command Language

This section describes how to use the DB2 PM command language.

Entering DB2 PM Commands

When entering DB2 PM commands, the command keyword identifies the start of a command. For example:

Example of a DB2 PM Command

```
⋮
ACCOUNTING
⋮
```

You can enter one occurrence of each report set command per command stream. You can enter a report set command by itself (as shown above) or you can enter the command, a subcommand, and various options for that subcommand. Subcommands identify the functions (REDUCE, REPORT, SAVE, RESTORE, TRACE, or FILE) to be processed for the command request.

Example of a DB2 PM Command

```
⋮
STATISTICS
     TRACE
       LAYOUT (LONG)
⋮
```

Subcommands and options do not need to be enclosed in parentheses.

Entering Auxiliary Commands and Troubleshooting Commands

Auxiliary commands have no subcommands. Each auxiliary command consists of the command and one or more options. Apart from this, report set commands and auxiliary commands are syntactically alike. All other syntax rules apply to both types of commands.

Example of a DB2 PM Auxiliary Command

```
⋮
GLOBAL
     INTERVAL (30)
     PAGESIZE (55)
     INPUTDD  (SMFILE)
⋮
```

You can only enter GLOBAL, DUMP, and TAPECOPY commands once in a command stream, but other auxiliary commands can be entered more than once. You can enter a command with an option and a parameter.

General Rules for Using the DB2 PM Command Language

This section describes the general rules for using the DB2 PM command language.

Value Formats

Certain values have a standard format. In general, the following applies:

Numeric Values
Can have up to 16 digits. Any exceptions or specific maximums are specified by command.

Name Values
These values can contain any combination of letters, numbers, and special characters, except for the delimiters blank, dash, parentheses, and comma.

DDNAMEs
Can be from 1 to 8 characters in valid JCL ddname format.

Dates
The default format is *mm/dd/yy*. This can be changed using the DATEFORMAT parameter. A new date begins at 00:00:00.00.

Times
The format is *hh:mm:ss.th* where *th* (tenths and hundredths of a second) is optional. Midnight is 00:00:00.00 of the next day. The latest time before midnight is 23:59:59.99. Trailing zeros are not required.

Entering Comments in the DB2 PM Command Language

Only command text entered between columns 1 and 72 is considered for processing. A command string can be continued anywhere on the following line, without continuation characters. All lines of the command string that are read are printed in the DPMLOG DD statement output. You can enter any number of continuation lines. Blank lines and comment lines are printed but are not considered for processing. Comments are entered within the comment delimiters:

/* (comment line begin)
*/ (comment line end)

or you can comment out an entire line by placing an asterisk (*) in column 1.

Appendix E. 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, 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
Informationssysteme GmbH
Department 3982
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

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

C/370	CICS
DATABASE 2	DB2
DRDA	Distributed Relational Database Architecture
GDDM	IBM
MVS	MVS/ESA
OS/2	OS/390
QMF	RACF
RMF	VTAM

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through The Open Group.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Other company, product or service names may be trademarks or service marks of others.

Glossary

The following terms and abbreviations are defined as they pertain to DB2 PM. This glossary does not include terms defined in the prerequisite DB2 manuals. If you do not find the term you are looking for, refer to the index, to the glossary of *IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide*, or to the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

A

accounting by DB2 PM identifier graphs. Graphs provided by IRF graphics. The graphs show plots representing values of selected DB2 PM identifiers.

accounting by field identifier graphs. Graphs provided by IRF graphics. The graphs show plots representing values of selected DB2 fields.

accounting data. Data that provides summary information about DB2 resource use. Accounting data is used to summarize DB2 local and distributed activity associated with the execution of specified application plans, detect potential problem areas within DB2, track resource use on a regular basis, and identify DB2 threads that fail to meet certain user-specified criteria.

accounting report set. A set of DB2 PM reports providing information about DB2 resource activity associated with DB2 applications.

accounting trace data. DB2 trace data that provides summary information about DB2 resource use.

ACE. A DB2 PM identifier denoting agent control element address.

administrator user. A TSO user with the authority to access the administrator dialog panels, where data collector parameters can be altered.

agent control element address (ACE). The absolute hexadecimal address of the DB2 agent control element for the thread. The ACE provides identification for each thread, and can be used to select records for a particular thread. Note that an ACE address can be reused after a thread terminates.

allied thread. A DB2 PM term for a thread that originates and terminates at the same location without requesting data from any other locations.

allied-distributed thread. A DB2 PM term for a thread that originates and terminates at the same location and that requests data from at least one other location.

audit report set. A set of DB2 PM reports that provide a means of tracking DB2 resource access. The audit reports include information on who performed an audited action, when it was performed, and what auditable DB2 objects were involved. The reports can be used to track DB2 resource access, identify potential security breaches and violations, and track granting and revoking of DB2 privileges.

audit trace data. DB2 trace data that identifies resource users, persons issuing use authorization to others, and possible security violations against audited DB2 tables.

AUTHID. A DB2 PM identifier denoting authorization ID.

AUTO command. Used to refresh and redisplay the current thread activity or statistics panel periodically.

auto display. A function that refreshes and redisplays the current thread activity or statistics panel periodically.

B

background processing. The execution of low-priority jobs when higher-priority programs are not using the system resources. In TSO, a background job is entered through the SUBMIT command or through SYSIN. Contrast with foreground processing.

BOUNDARY option. An option of the GLOBAL command and the REDUCE subcommand that specifies the starting time of the intervals used to summarize records in the reduction process, by indicating a number of minutes past the hour.

C

CLASS. A DB2 PM identifier denoting trace class.

CODES option. An option of the DUMP and TAPECOPY commands used to define code values for records to be processed.

COLLECT command. Used to access the collect report data facility.

collect report data. An option used to start one or more DB2 traces, directing the resultant trace data to a data set.

collect task. A task that has been configured to start and stop DB2 traces for collecting specific DB2 instrumentation data and directing this data to a data set. Up to four independent collect tasks can be configured and started using the collect report data facility.

CONNECT. See *connection ID*.

connection ID (CONNECT). A DB2 PM identifier derived from the connection ID of the correlation header. This is the type of subsystem used to interface with DB2, such as CICS, TSO, or IMS.

connection type (CONNTYPE). An identifier showing the type of connection for an associated thread.

CONNTYPE. See *connection type*.

CONVERT. A DB2 PM function to change SAVE data sets into sequential data sets suitable for use by the DB2 Load Utility. See also *save-file utility*.

correlation ID. An identifier consisting of two fields: the correlation name and the correlation number. It is used primarily by CICS and IMS. For some connection types the correlation number is blank. DB2 PM defaults to an 8-byte correlation name, followed by a 4-byte correlation number. You can specify a different translation using the CORRDATA member of the DPMPARMS data set. See also *correlation name*, *correlation number*, and *correlation translation*.

correlation name (CORRNAME). An identifier assigned to a task. This value is a subset of the correlation ID. Its meaning varies with the connection type. See also *correlation ID*.

correlation number (CORRNMBR). An identifier assigned to a task. This field is a subset of the correlation ID. Its meaning varies with the connection type. See also *correlation ID*.

correlation translation. A DB2 PM function that interprets the information contained in the correlation ID. See *correlation ID*, *correlation name*, *correlation number*, and *correlation translation data set*.

correlation translation data set. A DB2 PM data set used to define and interpret the use of the correlation ID for processing within DB2 PM reports and panels. See also *correlation ID*.

CORRNAME. See *correlation ID*.

CORRNMBR. See *correlation ID*.

D

DATABASE. A DB2 PM identifier denoting the name of the DB2 database.

database access thread (DBAT). A DB2 PM term for a thread that performs work at the request of an allied-distributed or DBAT-distributed thread originating at another location.

database identifier (DBID). A decimal identifier of a database.

DATASET. A DB2 PM identifier consisting of the 8-character name of the active log, the archive log, or the bootstrap data set.

DBAT. A DB2 PM identifier denoting database access thread.

DBAT-distributed. A DB2 PM term for a DBAT that requests work from another location.

DB2 instrumentation facility. A DB2 facility that controls and collects DB2 monitoring, auditing, performance, accounting, statistics, and serviceability (global) data. Synonymous with *DB2 trace facility*.

DB2 location. The name assigned to the DB2 subsystem. This name is unique within a set of subsystems that can communicate with each other. See also *local location*.

DB2 trace facility. Synonym for *DB2 instrumentation facility*.

DB2ID. A DB2 PM identifier denoting DB2 subsystem ID. Synonymous with *SUBSYSTEMID*.

DB2 PM command language. A set of commands and subcommands used to control DB2 PM processing.

DB2 PM identifier. A DB2 PM term for an identifier that specifies various attributes of the object upon which DB2 PM is reporting. Used by DB2 PM to group data, order reports, and include or exclude specific data.

DB2 PM logs. Reports that provide summarized information about events during DB2 PM execution. See *exception log*, *message log*, *job summary log*, *IFCID frequency distribution log*, and *DPMLOG*.

DDNAME option. An option specifying the DDNAME of a data set. With REPORT and TRACE subcommands, the option is used to specify the data set where the report or trace is written; with the SAVE and FILE subcommands, it is used to specify the destination data set to which the save data is written; with RESTORE, it is used to specify the data set from which restored data is read.

DELTA command. Used to invoke delta processing mode for viewing system-wide statistics in time slices.

delta processing mode. A statistics processing mode that reports field values in time slices.

delta record. A data structure describing the activity of a DB2 subsystem over a specific period of time.

delta value. A counter value found in a DB2 PM statistics delta record representing activity in a DB2 system over the time period described by the delta record.

display exception list. A list that contains a brief description of each exception condition that was detected by the display exception processor.

display exception processing. An Online Monitor subtask used to check for exception conditions while viewing DB2 activity on the screen.

DISTDD option. An option of the GLOBAL command used to specify the DDNAME for the frequency distribution data set.

DISTRIBUTE command. An auxiliary command used to calculate frequency distributions for selected fields. The results can be viewed or printed using the graphics function of the IRF.

DPMLOG. A DB2 PM log that contains messages about DB2 PM processing.

DPMOUT. The DB2 PM output data set. DB2 PM generates the output data using SMF or GTF data as input. The DPMOUT data set contains data converted to a standardized format, sorted, and written to a format that can be used to create DB2 PM reports, traces, and data sets.

DPMPARMS. A DB2 PM data set that contains information about customized functions.

Dump Name option. An option of the DUMP command used to specify a name to be printed on the dump.

DUMP processing. A DB2 PM function used to provide a printout of an input data set in dump format. All records on the input data set, a selected range of records, or specific record types can be used.

E

exception DPMOUT data set. A data set containing DB2 instrumentation records (in DPMOUT format) with at least one field in exception status. The data set can be used as input to the Batch record trace or statistics trace.

exception event processing. An Online Monitor subtask used to monitor the DB2 subsystem for the presence of particular events.

exception log. A log presenting a list that identifies DB2 accounting and statistics records with at least one field outside user-specified limits.

exception log file data set. A sequential data set suitable for use by the DB2 load utility. The data set contains a listing of accounting and statistics exception records identical to the listing in the exception log. See *exception log*.

exception processing. Reporting that provides a means to identify DB2 threads and statistics intervals with fields that contain exceptional values. It is used to highlight problems in the DB2 subsystem and in threads causing performance problems. Exception processing is performed against accounting and statistics instrumentation data.

exception profiling. A DB2 PM function that automatically sets exception threshold values to the exception threshold data set. The values are based on user-specified criteria and instrumentation data from a particular DB2 site.

exception reports and traces. Reports and traces that are obtained using the EXCEPTION option of the accounting or statistics REPORT or TRACE subcommands. Exception reports and traces can have different layouts. See also *exception processing*.

exception threshold data set. A data set used to define exception thresholds for statistics and accounting field IDs.

EXEC command. A command that instructs DB2 PM to execute the commands in a job stream.

EXEC option. An option of the DUMP and TAPECOPY commands used to define the DDNAME of the input data set.

EXPLAIN command. Used to invoke the explain function. Explain is a function that describes the access paths and processing methods chosen by DB2 for a given SQL statement.

explain report set. A set of DB2 PM reports containing access path information and catalog information.

F

FIELD. A DB2 PM identifier for a field in the instrumentation record and comparison value, as defined using the FIELD command.

FIELD command. An auxiliary command used to locate exception conditions for individual DB2 records. The command can only be used with the record trace INCLUDE/EXCLUDE options.

FIELDID option. An option of the DISTRIBUTE command used to specify the keyword for the field.

FILE processing. A DB2 PM function used to format accounting, audit, and statistics data and store it in sequential data sets suitable for use with the DB2 load utility.

foreground. In TSO, the environment in which programs are swapped in and out of main storage so that terminal users can share processing time. Contrast with background.

frequency distribution data set. Distribution data set generated as a result of the DISTRIBUTE command.

frequency distribution graph. A graph provided by the IRF graphics. It shows the distribution of a given field's data over user-defined ranges.

FROM/TO option. An option that defines the range of record timestamps processed by DB2 PM. FROM specifies the starting date and time; TO specifies the finishing date and time. The option can be used with the GLOBAL command and the REDUCE, REPORT, TRACE, and FILE subcommands.

G

GLOBAL command. An auxiliary command used as a primary filter for input records, and to specify default values for parameters within the other DB2 PM subcommands.

global parameters function. Used to set the DB2 subsystem ID and the plan name for an Online Monitor session, specify output data sets, and establish defaults for the autodisplay function.

graphics processing. A capability provided by DB2 PM that allows creation of graphs representing statistics, accounting, or frequency distribution information based on previously reduced data. Graphics are part of the Interactive Report Facility.

GROUP. A DB2 PM identifier that specifies the name of a Data Sharing Group.

GROUP command. An auxiliary command that can be used to specify up to ten values of one DB2 PM identifier type.

H

high water mark. A status value found in a DB2 trace record representing the highest value of some aspect of a DB2 system over a period starting when DB2 was started and ending at the timestamp of the trace record.

HISTORY command. Used to recall and view thread activity, statistics, and system parameters data (IFC records) previously collected by the data collector.

I

I/O activity report set. A set of DB2 PM reports reflecting reads, writes, and other I/O activities performed throughout the DB2 subsystem. The reports are used to track I/O volumes and service times and to summarize I/O activity for the active log, archive log, BSDS, the buffer pool manager, and the EDM pool manager.

ICU. Interactive Chart Utility.

IDENTIFIERS option. An option of the DISTRIBUTE command used to specify the DB2 PM identifiers used in collecting distribution data.

IFCID. A DB2 PM identifier denoting instrumentation facility component identifier.

IFCID frequency distribution log. A DB2 log presenting a count of input and processed records by IFCID, and the percentage of the total number of trace records that each IFCID represents.

INCLUDE/EXCLUDE option. An option used to select records from an input data set. INCLUDE is used to define a list of DB2 PM identifier values that are included in DB2 PM processing, and EXCLUDE to define a list of DB2 PM identifier values that are excluded from processing. The option is available with the GLOBAL command and the REDUCE, REPORT, TRACE, and FILE subcommands.

INPUTDD option. An option of the GLOBAL command used to specify the DDNAME of the input data set.

INSTANCE. A DB2 PM identifier that is part of the LUWID for distributed activity, and can be used to match the activity performed by DBATs, DBAT-distributed threads, and allied-distributed threads.

instance number. A DB2 PM identifier consisting of a unique number assigned to a thread involved in distributed activity. It is used to match an allied-distributed thread to its database access threads.

Interactive Chart Utility (ICU). A charting utility of the graphical data display manager (GDDM).

Interactive Report Facility (IRF). A DB2 PM facility that allows request for reports through a set of menus and interactive panels. The IRF also creates and submits control statements for background generation of DB2 PM reports.

INTERVAL command. Used to invoke Interval mode for viewing an accumulation of statistics data from a specified time.

INTERVAL identifier. A DB2 PM identifier used in accounting and statistics report sets to order data according to the reduction intervals.

INTERVAL option. An option of the GLOBAL command and the REDUCE subcommand that defines the time interval for consolidating records.

interval processing mode. The statistics processing mode that displays an accumulation of statistics data from a specified time.

IRF. Interactive Report Facility.

J

job summary log. A DB2 PM log providing a summary of events during DB2 PM execution. It provides information about the detection of DB2 START/STOP commands, report set reduction interval completion, SAVE and RESTORE subcommand completion by report set, and major errors and messages.

L

LAYOUT option. An option of the accounting and statistics REPORT and TRACE subcommands used to specify the layout of the report and the amount of detail.

LEVEL option. An option of the REPORT, TRACE, and FILE subcommands used to specify the type of report or trace, and the amount of detail.

LIMITS option. An option of the DISTRIBUTE command used to specify the limits of the ten ranges for frequency distribution.

LIST command. An auxiliary command used to define a named list of DB2 PM identifier values. A list can contain up to ten members of one DB2 PM identifier type.

local location. The location that is the focus of the report (the location indicated in the report header). See also *LOCATION*.

LOCATION. A DB2 PM identifier denoting the name assigned to a subsystem. See also *local location*.

locking report set. A set of DB2 PM reports and traces providing lock suspension, lock timeout, and deadlock information.

lockout. DB2 PM uses this term as an aggregate name for both timeout and deadlock.

LOOK command. Used to invoke the Look Selections menu, or to select certain exception processing and trace facility functions.

M

MAINPACK. A DB2 PM identifier used to identify the first or last package or DBRM in a plan. MAINPACK is used in the accounting report set.

MAXDUMP option. An option of the DUMP command used to define the length of dump in bytes.

MEMBER. A DB2 PM identifier that specifies the name of a member in a Data Sharing Group.

merged report. Reports and traces that include, in the same report, the local and distributed activity associated with each thread. Information is provided for allied threads created at the local site, allied-distributed threads created at the local site, and DBATs performed remotely on behalf of allied-distributed threads created at the local site. Merged reports are available in the accounting and SQL activity report sets.

message log. A DB2 PM log that lists messages about DB2 PM internal sort.

MIGRATE. A DB2 PM function to change SAVE data sets produced by previous releases into the current release format. See also *save-file utility*.

monitor data. DB2 trace data that provides thread, statistical, and performance DB2 data to the Online Monitor component of DB2 PM.

multi-site report. Report or trace that contains data from multiple DB2 locations. Multi-site reports can be nondistributed (when no interaction takes place between the DB2 subsystems reported) or distributed (when interaction occurs between DB2 subsystems). Multi-site reports are the default in all report sets. See also *single-site report*.

N

N/A. Not applicable.

N/C. Not calculated.

N/P. Not present.

NAME option. An option of the TAPECOPY command used to define a name identifying the records copied.

NEWCOPY option. An option of the TAPECOPY command used to define the DDNAME of the output data set.

nonmerged report. Reports and traces that separate activity according to the location where the activity occurred. All threads performed at each location are reported together.

O

OFFSET option. An option of the DUMP and TAPECOPY commands used to define the offset of the record code.

Online Monitor. The online monitoring part of DB2 PM used to monitor an active DB2 subsystem.

OPTIONS command. This command can either be used from an Online Monitor panel to set various options controlling the environment of your IRF session, or from the source explain SQL Statement Selection window to change various source explain processing options.

ORDER option. An option of the REPORT and TRACE subcommands. ORDER is used to specify the DB2 PM identifiers and their sequence, in sorting reported information.

ORIGAUTH. A DB2 PM identifier denoting original authorization ID.

original authorization ID (ORIGAUTH). A DB2 PM identifier consisting of the original value of the Primary Authorization ID when a thread was created (before it could be changed by any authorization exits).

P

PACKAGE. A DB2 PM identifier used to identify a package, regardless of the plan it belongs to. PACKAGE is used in the accounting report set.

PAGESET. A DB2 PM identifier denoting the name of a page set. A page set consists of a collection of tables or an index.

PAGESIZE option. An option of the GLOBAL command used to specify the number of printed lines per page.

performance trace data. DB2 trace data about various subsystem events that are used for program, resource, user, and subsystem-related tuning.

periodic exception list. A list that contains a brief description of each exception condition that was detected by the periodic exception processor.

periodic exception processing. An Online Monitor subtask that gathers DB2 thread activity and statistics information at user-specified intervals and analyzes the data for exception conditions.

PLANNAME. A DB2 PM identifier denoting plan name.

PRESORTED option. An option of the GLOBAL command used to specify whether DB2 PM internal sort is used.

PRIMAUTH. A DB2 PM identifier denoting primary authorization ID.

Q

QUALIFY command. Used to filter the threads listed on the Thread Summary panel.

R

record trace report set. Reports that format each DB2 trace record selected. This report set can be used to format a subset of data obtained from DB2 statistics, accounting, or performance trace information. The reports also provide access to users who need to view DB2 serviceability fields.

REDUCE processing. A DB2 PM function used to decrease the volume of the input data to subsequent REPORT and SAVE processing. REDUCE consolidates individual records with certain common characteristics into one record for reporting purposes.

regular mode. Statistics processing mode that displays values accumulated since DB2 startup.

REINIT command. A command used to restart the Exception Processor using the latest values in the Exception Threshold data set.

remote location. A location requesting or providing data for the local location.

report. A listing that uses summarized data. In reports events are ordered by DB2 PM identifiers, such as authorization ID or plan name.

report set. One or more DB2 PM reports containing the same information in varying degrees of detail. Report set names indicate the nature of the information included. For example, the statistics report set addresses DB2 system-wide data and the I/O activity report set addresses I/O activity.

REQLOC. A DB2 PM identifier denoting requesting location.

request type (REQUESTTYPE). A DB2 PM identifier used to identify a locking-related request.

requesting location (REQLOC). If the thread is an allied thread or the thread is an allied-distributed thread (this location is the requester), the requesting location is the same as the local location. If the thread is a database access thread (this location is a server), the requesting location is the location that made the request.

REQUESTTYPE. A DB2 PM identifier denoting request type.

RESET command. Used to reset the QUALIFY, SORT, DELTA, and INTERVAL commands.

resource manager ID (RMID). The decimal identifier of a DB2 Resource Manager.

resource type (RESOURCETYPE). A DB2 PM identifier used to identify a locked resource.

RESOURCETYPE. A DB2 PM identifier denoting resource type.

RESTORE processing. A DB2 PM function used to reload a previously saved data set for additional processing. To restore reduced data, DB2 PM takes the saved, reduced data out of the external data sets. Next, DB2 PM places that data in main storage in the same reduced format as it was before it was saved. It is possible to restore previously reduced data, subsequently reduce new raw data, and use both for reporting or additional saving.

RID list. Record identifier list.

RMID. A DB2 PM identifier denoting resource manager ID.

S

SAVE processing. A DB2 PM function used to produce a data set containing reduced records. To save reduced data, DB2 PM copies the reduced data from main storage to external data sets. Save is available in accounting and statistics.

save-file utility. A utility which consists of the MIGRATE and CONVERT functions: to migrate save data sets from one DB2 PM version to another (MIGRATE) and to change SAVE data sets into sequential data sets suitable for use by the DB2 load utility (CONVERT). The save-file utility is used only with accounting and statistics save data sets.

serving location. The name of a DB2 location that performs work on behalf of another (requesting) location. See also *LOCATION*.

single-site report. A single-site report includes data from a single DB2 subsystem only. It is obtained by processing an input data set containing data from a single site or by specifying a single location using INCLUDE/EXCLUDE. See also *multi-site report*.

SKIP option. An option of the DUMP and TAPECOPY commands used to define the number of records to be skipped before processing begins.

snapshot. A status value found in a DB2 trace record representing the status of some aspect of a DB2 system. Snapshot values are also found in DB2 PM statistics delta records. In this case they represent the status of the DB2 system current at the end time of the period described by the delta record.

SORT command. Used to order the threads displayed on the Thread detail panel by one or more field values.

SORTBY. An option of the SQL activity REPORT and TRACE subcommands used to sort SQL events within each summary level.

source explain. A facility used to explain SQL statements that are embedded in a source program or SPUFI input from within an ISPF/PDF editor.

SQL activity report set. A set of reports that provides information on SQL activity taking place during the processing of a DB2 application. The reports are used to analyze SQL calls within a logical unit of work, analyze the access path selected by the DB2 optimizer for local SQL calls, observe the local Data Manager scans, locking activity, and Buffer Manager I/O activity taking place during the execution of SQL requests, and group SQL activities by logical units of work.

statistics graph. A graph provided by the IRF graphics. It shows plots representing values of collected statistics data.

statistics processing. A function of DB2 PM that provides information about the current activity over the entire DB2 subsystem. The data is logged at intervals, specified at DB2 installation. The panels are used to view system-wide statistics for key DB2 components and assess system-wide performance. Details of DB2 statistics fields, buffer pools, and distributed activity can also be viewed. See also *exception processing*.

statistics report set. A report set that provides information collected by the DB2 Instrumentation Facility over the entire DB2 subsystem. The data is logged at intervals specified at DB2 installation. The reports are used to view system-wide statistics for key DB2 components, compare system performance in several reporting intervals, and assess system-wide performance and summarize it in one report. See also *exception processing*.

statistics trace data. DB2 data used for DB2 capacity planning and tuning the entire set of DB2 programs.

STOPAFT option. An option of the DUMP and TAPECOPY commands used to define the number of records to be processed, starting after the number of records to be skipped. See also *SKIP option*.

SUBSYSTEMID. A DB2 PM identifier denoting DB2 subsystem ID. Synonymous with *DB2ID*.

SUMMARIZEBY. An option of the SQL activity REPORT and TRACE subcommands used to summarize SQL events.

SYSPRMDD option. An option of the GLOBAL command used to specify the DDNAME for the system parameters report. See also *system parameters report set*.

system parameters. An option that displays the current DSNZPARM values of a DB2 subsystem. It displays user-assigned control values of key DB2 installation values.

system parameters report set. A report set that shows current values of key DB2 installation values. It shows user-assigned control values of key DB2 installation values. See also *SYSPRMDD option*.

T

TAPECOPY processing. A DB2 PM function used to make a copy of all or part of an SMF input data set.

thread activity. A function that provides information about the current activity of all active threads connected to a DB2 subsystem.

thread type (THREADTYPE). A DB2 PM identifier used to identify the type of thread.

THREADTYPE. A DB2 PM identifier denoting thread type.

time zone information member. A member of the DPMPARMS data set that contains user-specified information about time zone processing.

timestamp. A value that consists of a date, a time, and a number of microseconds.

TIMEZONE option. An option of the GLOBAL command used to specify the time difference between DB2 locations.

TOP option. An option in the accounting report set that can be used to produce a list of applications or users that have required most use of specific DB2 resources.

trace. A listing that uses unreduced data and usually reports individual records. A trace generally produces output in a short or long format; these are similar to the report summary and detail levels.

U

UNITS option. An option of the DISTRIBUTE command used to specify the units for the LIMITS option. See also *LIMITS option.*

user-tailored reporting. A DB2 PM function that allows the users to tailor accounting and statistics report and trace layouts. The tailoring is performed using the IRF.

utility activity report set. A set of DB2 PM reports providing DB2 application work information categorized as bind processing and utilities. These reports are used to determine workload distribution for local and distributed activity and to associate DB2 work with individual users or DB2 PM identifiers.

W

WORKLOAD. An option of the SQL activity REPORT and TRACE subcommands used to display workload detail for SQL events.

Bibliography

IBM DB2 Performance Monitor for OS/390 Version 6 Report Reference Volume 1, SC26-9164

IBM DB2 Performance Monitor for OS/390 Version 6 Report Reference Volume 2, SC26-9165

IBM DB2 Performance Monitor for OS/390 Version 6 Online Monitor User's Guide, SC26-9168

IBM DB2 Performance Monitor for OS/390 Version 6 Batch User's Guide, SC26-9167

IBM DB2 Performance Monitor for OS/390 Version 6 Command Reference, SC26-9166

IBM DB2 Performance Monitor for OS/390 Version 6 Messages, SC26-9169

IBM DB2 Performance Monitor for OS/390 Version 6 Using the Workstation Online Monitor, SC26-9170

IBM DB2 Performance Monitor for OS/390 Version 6 Installation and Customization, SC26-9171

IBM DB2 Performance Monitor for OS/390 Version 6 General Information, GC26-9172

IBM DB2 Performance Monitor for OS/390 Version 6 Data Collector Application Programming Interface Guide, SC26-9173

Program Directory for IBM DB2 UDB Server for OS/390 DB2 Performance Monitor DB2 Workstation Analysis and Tuning Version 6, GI10-8183

IBM DB2 Universal Database Server for OS/390 Version 6 Administration Guide, SC26-9003

IBM DB2 Universal Database Server for OS/390 Version 6 Command Reference, SC26-9006

IBM DB2 Universal Database Server for OS/390 Version 6 SQL Reference, SC26-9014

IBM DB2 Universal Database Server for OS/390 Version 6 Application Programming and SQL Guide, SC26-9004

IBM DB2 Universal Database Server for OS/390 Version 6 Data Sharing: Planning and Administration, SC26-9007

IBM DB2 Universal Database Server for OS/390 Version 6 Installation Guide, GC26-9008

IBM DB2 Universal Database Server for OS/390 Version 6 Utility Guide and Reference, SC26-9015

IBM DB2 Universal Database Server for OS/390 Version 6 Diagnosis Guide and Reference, LY36-3736

IBM DB2 Universal Database Server for OS/390 Version 6 Messages and Codes, GC26-9011

DSFSMS/MVS Macro Instructions for Data Sets, SC26-4913

OS/390 MVS System Management Facilities (SMF), GC28-1783

DFSMS/MVS Access Method Services for ICF, SC26-4906

DFSMS/MVS Access Method Services for VSAM, SC26-4905

OS/390 MVS Initialization and Tuning Guide, SC28-1751

OS/390 MVS System Codes, GC28-1780

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 Writing Transaction Programs for APPC/MVS, GC28-1775

OS/390 ISPF Dialog Developer's Guide and Reference, SC28-1273

OS/390 ISPF Services Guide, SC28-1272

OS/390 TSO/E Messages, GC28-1978

VTAM for MVS/ESA Resource Definition Reference, SC31-6552

OS/390 TSO/E REXX Reference, SC28-1975

TCP/IP for MVS: Application Programming Interface Reference, SC31-7187

TCP/IP Tutorial and Technical Overview, GG24-3376

ITSC - A Beginner's Guide to MVS TCP/IP Socket Programming, GG24-2561

Graphic Data Display Manager/Presentation Graphics Feature: Interactive Chart Facility User's Guide, SC33-0111

IBM Dictionary of Computing, New York: McGraw-Hill, 1994

Index

D

E

F

G

M

N

O

P

Q

R

S

T

U

V

W

Readers' Comments — We'd Like to Hear from You

DB2 Performance Monitor for OS/390
Batch User's Guide

Publication No. SC26-9167-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC26-9167-01

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Entwicklung GmbH
Information Development, Dept. 0446
Schoenaicher Strasse 220
71032 Boeblingen
Germany

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Program Number: 5645-DB2

Printed in the United States of America

SC26-9167-01