

IBM Tivoli OMEGAMON XE for DB2 Performance Expert on
z/OS
IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on
z/OS
Version 5.3.0

Reporting User's Guide



IBM Tivoli OMEGAMON XE for DB2 Performance Expert on
z/OS
IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on
z/OS
Version 5.3.0

Reporting User's Guide



Note

Before using this information and the product it supports, read the information in “Notices” on page 301.

Second edition, September 2015

This edition applies to the following releases and to all subsequent releases and modifications until otherwise indicated in new editions:

- IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS, version 5, release 3, modification 0 (5655-W37)
- IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS, version 5, release 3, modification 0 (5655-W38)

This edition replaces SH12-7053-00.

© **Copyright IBM Corporation 1985, 2015.**

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

Contents

About this information	ix
Conventions used in the OMEGAMON documentation	ix
Terminology used	xii
How to read syntax diagrams	xii
Where to find information	xiv
Service updates and support information	xv
Accessibility features	xv
How to send your comments	xv
 What's new	 xvii
 Chapter 1. Introduction to OMEGAMON XE for DB2 PE	 1
 Chapter 2. Introduction to batch reporting	 3
What OMEGAMON XE for DB2 PE does	3
Output types	4
Reports and traces	4
Data sets	4
Logs	4
Report sets	4
Accounting reports and traces	5
Statistics reports and traces	5
SQL Activity reports and traces	6
Locking reports and traces.	6
I/O Activity reports	6
Audit reports and traces	6
Utility Activity reports and traces	6
Record Trace reports.	7
Explain reports	7
System Parameters reports.	7
Levels of detail of report sets	7
Input for reports	7
Overview of the reporting process	8
 Chapter 3. Collecting data for reports	 9
Managing DB2 traces with the Workstation Online Monitor	10
Collecting report data with the ISPF Online Monitor.	10
Accessing the Collect Report Data (CRD) panels	11
Configuring a collect task.	13
Restricting data types for selected report sets	14
Restricting IFCIDs for selected report sets	15
Restricting identifiers for selected report sets	16
Starting and stopping traces	17
Displaying trace status and messages.	19
DB2 trace termination	20
Collecting report data by specifying DB2 startup parameters	20
Collecting report data with the DB2 START TRACE command	20
START TRACE command examples	23
DB2 instrumentation data	24
Collecting report data with the Collect Report Data Batch program.	27
Collecting report data with the Near-Term History Data Collector	30
 Chapter 4. Creating reports using commands	 31
Setup information for data sets	31

ddnames of general data sets	31
ddnames of report data sets	33
General filters and controls	34
Report set commands	34
Subcommands and subcommand options	35
Example of a command string	36
Chapter 5. Creating reports with the Interactive Report Facility	39
Selecting reports and reports details	39
Specifying default options by using the OPTIONS command	41
Composing and executing the batch report command stream	41
Saving and recalling selections	42
Example of producing an Accounting report	42
Chapter 6. Using Statistics traces to get system overview information	49
Collecting Statistics data with a DB2 trace	49
Creating a Statistics trace report	50
Interpreting the Statistics trace report and applying rules of thumb	50
Getting an overview of your DB2 system performance	51
Buffer pool efficiency	52
EDM pool failures	53
Authorization problems	53
RID List failures	53
Locking problems	54
Logging performance	54
Data set Open and Close activities	54
ROWID efficiency	54
Thread management	54
Chapter 7. Using an Accounting report to analyze resource activities.	57
Collecting Accounting data with a DB2 trace	57
Creating a short Accounting report	58
Interpreting the Accounting report and making changes	58
Interpreting the Accounting trace report and applying rules of thumb	60
The number of Getpage requests per SQL statement	61
Comparison of class 1 and class 2 times	61
Comparison of class 2 elapsed and class 2 CPU times	61
Class 2 CPU time per SQL DML statement execution	61
Not accounted time	62
Class 3 suspensions	62
Chapter 8. Using Statistics and Accounting reports to identify exceptions	65
Output from exception reporting	65
Specifying exceptions using the Exception Threshold data set editor	66
How DB2 instrumentation data matches exception thresholds	70
Example of producing an Accounting exception report	71
Which exception fields and threshold values to choose	73
Chapter 9. Using SQL Activity reports to monitor SQL performance	75
When to use SQL Activity reports	76
Input for SQL Activity reports	77
Collecting SQL Activity data with a DB2 trace	79
Creating an SQL Activity trace	79
Interpreting the result	82
Remark about lengths of SQL statement texts in SQL Activity traces	82
Chapter 10. Using Explain reports to monitor access paths	85
Explain reports	85
Levels of detail	86

Required authorization	87
When to use Explain reports.	87
Creating an Explain report	88
Explain report example	88
Interpreting the Explain report	93
Chapter 11. Locking Activity reports and traces	97
When to use Locking reports	98
Input for Locking reports.	98
Collecting Locking Activity data with a DB2 trace	99
Creating a Detail Locking trace.	99
Chapter 12. I/O Activity reports.	103
When to use I/O Activity reports	103
Input for I/O Activity reports	104
Collecting I/O Activity data with a DB2 trace	104
Creating an I/O Activity trace.	104
Chapter 13. Utility Activity reports	107
When to run Utility Activity reports.	108
Input for Utility Activity reports	108
Collecting Utility Activity data with a DB2 trace.	109
Creating a Utility Activity trace	110
Chapter 14. Record Trace reports.	113
When to use Record Trace reports	113
Input for Record Trace reports.	113
Collecting Record Trace data with a DB2 trace	119
Creating a Record Trace report	119
Chapter 15. Housekeeping reports	121
The Audit report set	121
When to use Audit reports	122
Collecting Audit data with a DB2 trace.	122
Creating an Audit trace	122
The System Parameters report.	124
When to use System Parameters reports	124
Collecting System Parameters data with a DB2 trace	124
Creating a System Parameters report	124
Chapter 16. General Accounting information	137
Functions and utilities of the Accounting report set.	138
Thread types	139
Distributed activity	142
Query parallelism considerations.	143
Timing information	145
Input for Accounting reports	148
DB2 traces used in Accounting	148
Accounting record generation	149
Missing data sections.	150
Chapter 17. Creating effective Accounting reports	151
Processing considerations	151
Choosing the right level of detail.	152
Filtering data	153
FROM and TO subcommand options	153
INCLUDE and EXCLUDE subcommand options	154
Suppressing internal sort	154
Grouping data	154

Grouping data using sets of identifiers	155
Grouping data using the connection type identifier	155
Grouping data using the thread type identifier	156
Grouping data by ordering according to thread types	156
Specifying intervals for long-term reporting	157
Specifying exception thresholds for specific fields	157
Using a DPMOUT data set	157
Saving reduced data	157
Exception processing	158
ORDER processing	160
Default ordering: ORDER (PRIMAUTH-PLANNAME)	160
Ordering by connection ID: ORDER (CONNECT)	161
Ordering by correlation ID: ORDER (CORRNAME)	162
Examples of ordering by Plan, Main Package, and Package	163
Example of ordering by activity name: ORDER (ACTNAME)	166
Comparison of Accounting report types based on the above sample workload.	168
The impact of the DB2 subsystem parameter ACCUMACC	169
Example of ordering by interval: ORDER (INTERVAL)	170
TOP processing	171
Examples of TOP processing	171
Reducing data	173
Examples of interval calculation	174
Processing intervals	175
How intervals are calculated	176
Examples of interval processing	176
Examples using REDUCE	177
Examples using REDUCE and REPORT	178
The effect of REDUCE on TRACE	182
The effect of REDUCE on FILE	184
Member-scope and group-scope reporting	185
Member-scope reports	185
Group-scope reports	187
Chapter 18. General Statistics information	189
Statistics terms	190
Delta and interval records	190
Delta records	191
Interval records	191
Input for Statistics reports	193
Functions and utilities of the Statistics report set	193
Controlling the level of detail in reports and traces	195
Member-scope and group-scope reporting	196
Member-scope reports and traces	196
Group-scope reports and traces	198
Exception processing	202
Headers used in Statistics	204
How values are reported	205
Chapter 19. Exception processing	207
Exception processing output types	207
Accounting and Statistics exception reports	208
Accounting and Statistics exception traces	208
Accounting and Statistics Exception File data sets	208
Exception log data set	208
Exception Log File data set	208
Exception Threshold data set	209
Exception profiling	210
Data sets involved in exception profiling	210
Exception profiling dialog	211
Exception profiling method	213

Exception profiling report	214
Chapter 20. Reporting distributed data	215
Selecting threads	216
Examples of distributed transactions	216
Distributed transaction between DB2 host systems	216
Distributed transaction between DB2 host systems and DB2 systems on other platforms	217
Distributed transaction involving a DBAT-distributed thread	218
Chapter 21. Reporting data sharing information	219
Monitoring individual members	219
Example of a member-scope Locking report	219
Monitoring entire groups	221
Example of a group-scope Locking report	221
Group-scope Statistics	222
Group-scope Accounting	223
Collecting input data for group-scope reports	223
Chapter 22. Streamlining OMEGAMON XE for DB2 PE processing.	225
Filter data	226
FROM and TO	226
Example of reporting peak hour performance of a day	226
Example of reporting peak hour performance during a week	227
Example of generating an additional locking report	227
INCLUDE and EXCLUDE	228
OMEGAMON XE for DB2 PE identifiers	228
Example of reporting the performance of a specific location	229
Example of suppressing a specific connection type	229
Example of reporting about a specific user ID and plan	229
Suppressing internal sort	230
Choose the right level of detail	230
Group data	231
Use lists	231
Be careful with INTERVAL	232
Specify only relevant exception thresholds.	232
Do you need a DPMOUT data set?	233
Do you want to save reduced data?	233
Limit statements to be explained	233
Chapter 23. Customizing OMEGAMON XE for DB2 PE functions	235
Tailoring report layouts	236
Example of tailoring an Accounting report	236
Selecting a model layout	237
Adding a block.	239
Replacing fields	241
Displaying the layout.	243
Saving the layout	244
Using the layout	245
Specifying time zones	245
Time Zone Data Editor	246
Examples of time zone processing	247
Example 1: Time zones	248
Example 2: CPU clock settings.	249
Example 3: Time zones and CPU clock settings	250
Correlation ID translation	251
The default translation	251
Changing the default translation	252
Defining the MAINPACK identifier	252
Specifying the MAINPACK (example)	255
Examples of using the MAINPACK and PACKAGE identifiers	256

Modifying exception field descriptors	258
Chapter 24. The Performance Database and the Performance Warehouse	261
Accounting tables	263
Samples	264
Accounting File	264
Accounting Save-File	265
Audit tables	265
Samples	266
Exceptions table	267
Samples	267
Locking table	267
Samples	268
Record Trace tables	268
Samples	269
Statistics tables	270
Samples	272
System parameters	273
Samples	273
Creating data for the Performance Database and the Performance Warehouse	273
Chapter 25. Generating input data for spreadsheets.	275
Chapter 26. Further reference information	285
Using GROUP to improve Save data set performance	285
Comparing original authorization IDs with primary authorization IDs	286
Fields affected by roll-up during distributed and parallel tasks	287
Using ACCOUNTING REDUCE subcommand options to balance performance and data set space	288
Accessibility	293
Troubleshooting empty reports	296
Notices	301
Trademarks	303
Bibliography.	305
Index	307

About this information

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

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

This information is closely related to:

- *Report Command Reference*, which describes the commands to create reports
- *Report Reference*, which shows and explains the reports

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

- *Tivoli OMEGAMON® XE for DB2 Performance Expert on z/OS®* (PDFs and Techdocs on DB2 Tools Product Page)
- *Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS* (PDFs and Techdocs on DB2 Tools Product Page)
- *Tivoli Documentation Central*

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

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

Who should read this information

This information is for IBM data server professionals who need to produce and interpret OMEGAMON XE for DB2 PE reports.

“Conventions used in the OMEGAMON documentation”

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

Conventions used in the OMEGAMON documentation

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

Panels and figures

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

Symbols

The following symbols might appear in command syntax:

Symbol	Usage
	The or symbol is used to denote a choice. You can use the argument on the left or the argument on the right. For example: YES NO In this example, you can specify YES or NO.
()	Denotes optional arguments. Arguments that are not enclosed in square brackets are required. For example: APPLDEST DEST (ALTDEST) In this example, DEST is a required argument and ALTDEST is optional.
{ }	Some documents use braces to denote mandatory arguments, or to group arguments for clarity. For example: COMPARE {workload} - REPORT={SUMMARY HISTOGRAM} In this example, the workload variable is mandatory. The REPORT keyword must be specified with a value of SUMMARY or HISTOGRAM.
—	Default values are underscored. For example: COPY infile outfile - [COMPRESS={YES NO}] In this example, the COMPRESS keyword is optional. If specified, the only valid values are YES or NO. If omitted, the default is YES.

Notation conventions

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

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

For example:

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

Typeface conventions

This information uses the following typeface conventions:

Bold

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

- Keywords and parameters in text

Italic

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

Monospaced

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

Significant elements

Recommendation

Provides guidance when more than one option is available.

Related reading

Refers you to other publications that contain relevant information.

Requirement

Identifies a condition that must be met to ensure that the product is functional.

Restriction

Identifies a restriction or limitation with this product or an associated procedure.

“Terminology used” on page xii

IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS can be considered as a functional subset of IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS. Therefore the abbreviation OMEGAMON XE for DB2 PE or DB2 PE is used for both products. If a distinction is required, OMEGAMON XE for DB2 PM or DB2 PM is used explicitly.

“How to read syntax diagrams” on page xii

The rules in this section apply to the syntax diagrams that are used in this publication.

“Where to find information” on page xiv

You can access the documentation in several ways.

“Service updates and support information” on page xv

You can access support information for IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS and IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS on the Support home website, or you can use the IBM Support Assistant.

“Accessibility features” on page xv

Accessibility features help people with a physical disability, such as restricted mobility or limited vision, or with other special needs, to use software products successfully. This Knowledge Center is developed to comply with the accessibility requirements of software products according to Section 508 of the Rehabilitation Act of the United States.

“How to send your comments” on page xv

Your feedback is important in helping to provide the most accurate and high-quality information.

Terminology used

IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS can be considered as a functional subset of IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS. Therefore the abbreviation OMEGAMON XE for DB2 PE or DB2 PE is used for both products. If a distinction is required, OMEGAMON XE for DB2 PM or DB2 PM is used explicitly.

The following table shows the products that are described in this publication and the short names with which they are referred to throughout this publication:

Table 1. Product names and their short names

Product name	Short name
IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS	OMEGAMON XE for DB2 PE or DB2 PE
IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS	OMEGAMON XE for DB2 PM or DB2 PM
IBM DB2 Buffer Pool Analyzer for z/OS or a particular subsystem	Buffer Pool Analyzer
IBM DB2 database for z/OS	DB2

- Performance Expert Client and Workstation Online Monitor designate the client component of DB2 PE.

The client component of DB2 PE also designates the end user interface of Performance Expert for Multiplatforms, Performance Expert for Workgroups, and DB2 PE.

- OMEGAMON Collector designates the server component of DB2 PE.

How to read syntax diagrams

The rules in this section apply to the syntax diagrams that are used in this publication.

Arrow symbols

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

- ▶▶— Two right arrows followed by a line indicate the beginning of a statement.
- ▶ One right arrow at the end of a line indicates that the statement syntax is continued on the next line.
- ▶— One right arrow followed by a line indicates that a statement is continued from the previous line.
- ▶◀ A line followed by a right arrow and a left error indicates the end of a statement.

Conventions

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

- Footnotes are shown by a number in parentheses, for example, (1).

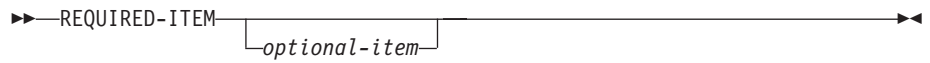
Required items

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



Optional items

Optional items appear below the main path.



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



Multiple required or optional items

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

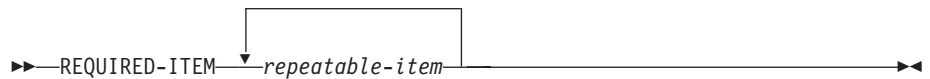


If choosing one of the items is optional, the entire stack appears below the main path.

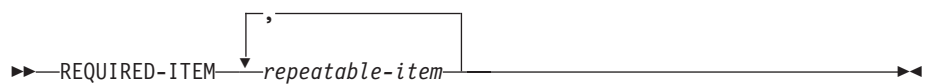


Repeatable items

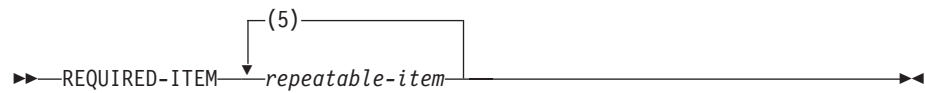
An arrow returning to the left above the main line indicates that an item can be repeated.



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



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



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

Default keywords

IBM-supplied default keywords appear above the main path, and the remaining choices are shown below the main path. In the parameter list following the syntax diagram, the default choices are underlined.



Where to find information

You can access the documentation in several ways.

The documentation for this product is provided in PDF and in HTML format at the following websites:

- *Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS*
- *Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS*

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software Knowledge Center website. You can access the Tivoli software Knowledge Center by going to the *Tivoli Documentation Central* website and clicking **O** under **Tivoli Documentation A-Z** to access all of the IBM Tivoli OMEGAMON product manuals.

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

The IBM Software Support website provides the latest information about known product limitations and workarounds in the form of technotes for your product. You can view this information at the *Support home* website.

Ordering publications

You can order many IBM publications such as product manuals or IBM Redbooks® online at the *IBM Publications Center* website.

You can also order by telephone by calling one of the following numbers:

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

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

Accessing terminology online

The *IBM Terminology* website consolidates the terminology from IBM product libraries in one convenient location.

Service updates and support information

You can access support information for IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS and IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS on the Support home website, or you can use the IBM Support Assistant.

Support home

On the *Support home* website, you can find service updates and support information including software fix packs, PTFs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads.

Accessibility features

Accessibility features help people with a physical disability, such as restricted mobility or limited vision, or with other special needs, to use software products successfully. This Knowledge Center is developed to comply with the accessibility requirements of software products according to Section 508 of the Rehabilitation Act of the United States.

The accessibility features in this Knowledge Center enable users to do the following tasks:

- Use assistive technologies, such as screen-reader software and digital speech synthesizer, to hear what is displayed on the screen. In this Knowledge Center, all information is provided in HTML format. Consult the product documentation of the assistive technology for details on using assistive technologies with HTML-based information.
- Operate specific or equivalent features using only the keyboard.
- Magnify what is displayed on the screen.

In addition, all images are provided with alternative text so that users with vision impairments can understand the contents of the images.

Navigating the interface by using the keyboard

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

Magnifying what is displayed on the screen

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

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information.

If you have any comments about this information or any other documentation, you can do one of the following actions:

- Complete and submit the *Reader Comment Form*.
- Send your comments by e-mail to swsdid@de.ibm.com.

Include the documentation name, the part number, the version number, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

What's new

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

SH12-7053-01 — September 2015

This edition replaces *IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS*; *IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS: Reporting User's Guide*, SH12-7053-00.

- Clarifications and corrections have been applied to the information where required.
- Information refresh for September 2015 - The following descriptions have been added or updated:
 - Several samples have been updated.
 - APAR number PI48187:
 - The new formats BP2C and GBP2C have been added to Chapter 25, “Generating input data for spreadsheets,” on page 275.
 - APAR number PI41417:
 - Update of “Record Trace tables” on page 268
 - Update of “Samples” on page 269
 - APAR number PI41760:
 - Update for new sample job FPEPCSVW and record types in Chapter 25, “Generating input data for spreadsheets,” on page 275.
 - Update of Simulated Buffer Pool records for Statistics:
 - “Statistics tables” on page 270
 - “Samples” on page 272
 - Chapter 25, “Generating input data for spreadsheets,” on page 275
 - APAR number PI23326: Update of “Accounting Save-File” on page 265.
- May 2015 - The following descriptions have been added or updated:
 - Several samples have been updated.
 - The DISTRIBUTE command has been removed from the information (“ddnames of general data sets” on page 31).
 - Update of the information in section “Interval records” on page 191.
 - Update of Deadlocks, Timeouts, Remote Locations:
 - Chapter 24, “The Performance Database and the Performance Warehouse,” on page 261
 - “Record Trace tables” on page 268
 - “Samples” on page 269
 - Chapter 25, “Generating input data for spreadsheets,” on page 275
 - Update of Aggregated Accounting records for Statistics:
 - “Statistics tables” on page 270
 - “Samples” on page 272
 - Chapter 25, “Generating input data for spreadsheets,” on page 275
 - The following information has been updated:

- "Example of ordering by activity name: ORDER (ACTNAME)" on page 166
- "Comparison of Accounting report types based on the above sample workload" on page 168

Chapter 1. Introduction to OMEGAMON XE for DB2 PE

OMEGAMON XE for DB2 PE is part of the integrated and complete cross zSeries monitoring solution of the IBM Tivoli OMEGAMON XE family, which monitors all DB2 subsystems on z/OS and other resources, such as IMS, MVS™, or CICS®.

For a comprehensive overview of the IBM Tivoli OMEGAMON XE family, its components and a graphical presentation of the component interrelations refer to the introductory section of *Configuration and Customization*.

Chapter 2. Introduction to batch reporting

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

You can use Performance Expert Client to:

- Examine a DB2 system and its applications in real time.
- Launch and control OMEGAMON XE for DB2 PE reporting functions.

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

“What OMEGAMON XE for DB2 PE does”

OMEGAMON XE for DB2 PE uses DB2 *instrumentation data* to generate performance reports in a form that is easy to understand and analyze.

“Input for reports” on page 7

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

“Overview of the reporting process” on page 8

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

What OMEGAMON XE for DB2 PE does

OMEGAMON XE for DB2 PE uses DB2 *instrumentation data* to generate performance reports in a form that is easy to understand and analyze.

You can use OMEGAMON XE for DB2 PE to:

- Determine DB2 subsystem performance and efficiency
- Identify and resolve potential problems
- Tune the DB2 subsystem
- Measure an application's performance and resource cost
- Tune applications and SQL queries
- Assess an application's affect on other applications and the system
- Gather information for cost purposes

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

“Output types” on page 4

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

“Report sets” on page 4

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

Output types

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

“Reports and traces”

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

“Data sets”

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

“Logs”

OMEGAMON XE for DB2 PE can log several activities and keep this information available in defined data sets.

Reports and traces

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

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

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

Data sets

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

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

Logs

OMEGAMON XE for DB2 PE can log several activities and keep this information available in defined data sets.

- *The Exception Log* contains Accounting and Statistics records with at least one field outside user-specified thresholds.
- *The IFCID Frequency Distribution Log* provides counts of input records by IFCID.
- *The Job Summary Log* provides a summary of events during OMEGAMON XE for DB2 PE execution.
- *The DPMLOG Execution Log* contains OMEGAMON XE for DB2 PE processing messages.

Report sets

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

“Accounting reports and traces” on page 5

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

“Statistics reports and traces” on page 5

Statistics reports and traces summarize information that is collected by the DB2

Instrumentation Facility for an entire DB2 subsystem. The collected data is logged at intervals that are specified when you install DB2.

“SQL Activity reports and traces” on page 6

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

“Locking reports and traces” on page 6

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

“I/O Activity reports” on page 6

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

“Audit reports and traces” on page 6

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

“Utility Activity reports and traces” on page 6

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

“Record Trace reports” on page 7

Record Trace reports show the contents of selected instrumentation records.

“Explain reports” on page 7

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

“System Parameters reports” on page 7

System Parameters reports show information about the configuration of your DB2 subsystem.

“Levels of detail of report sets” on page 7

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

Accounting reports and traces

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

Use this report set to:

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

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

Statistics reports and traces

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

Use this report set to:

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

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

SQL Activity reports and traces

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

Use this report set to:

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

Locking reports and traces

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

Use this report set to obtain:

- Summarized information about user activities related to lock suspensions and lockouts.
- Information about every lock request made by DB2.
- Information about lock details.
- Information about every suspension, lockout, detail, deadlock, or timeout during a specified interval.

I/O Activity reports

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

Use this report set to:

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

Audit reports and traces

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

Use this report set to:

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

Utility Activity reports and traces

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

Use this report set to:

- Analyze bind and utility activity within a logical unit of work.

- Associate DB2 work with individual users or OMEGAMON XE for DB2 PE identifiers.
- Recognize long utility or bind times, which can help you to identify bottlenecks in the system.

Record Trace reports

Record Trace reports show the contents of selected instrumentation records.

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

Explain reports

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

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

System Parameters reports

System Parameters reports show information about the configuration of your DB2 subsystem.

The report shows values for all DB2 system parameters that were in effect at the time the performance data was collected.

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

Levels of detail of report sets

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

The level of detail and the amount of data varies by report set and output type, as shown in Figure 1. Generally, start any problem determination task with high-level summary reports, such as Accounting and Statistics reports. Use traces if you require information about individual DB2 events.

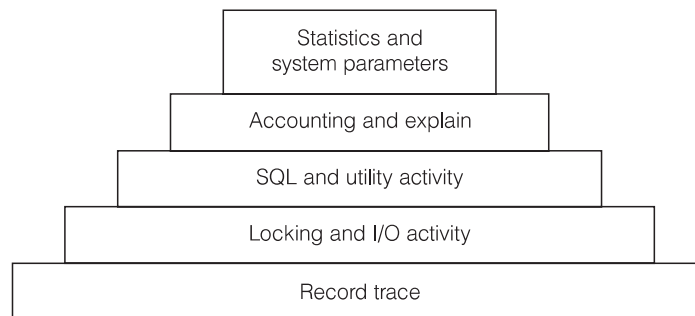


Figure 1. Report sets in order of detail

Input for reports

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

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

Overview of the reporting process

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

Figure 2 summarizes the main elements and functions involved in OMEGAMON XE for DB2 PE processing. The top part of the figure shows the various inputs to OMEGAMON XE for DB2 PE and the bottom part of the figure shows the different output types.

Solid lines indicate input or output, and broken lines indicate specifications the user makes. Words written in uppercase next to the arrows are subcommands.

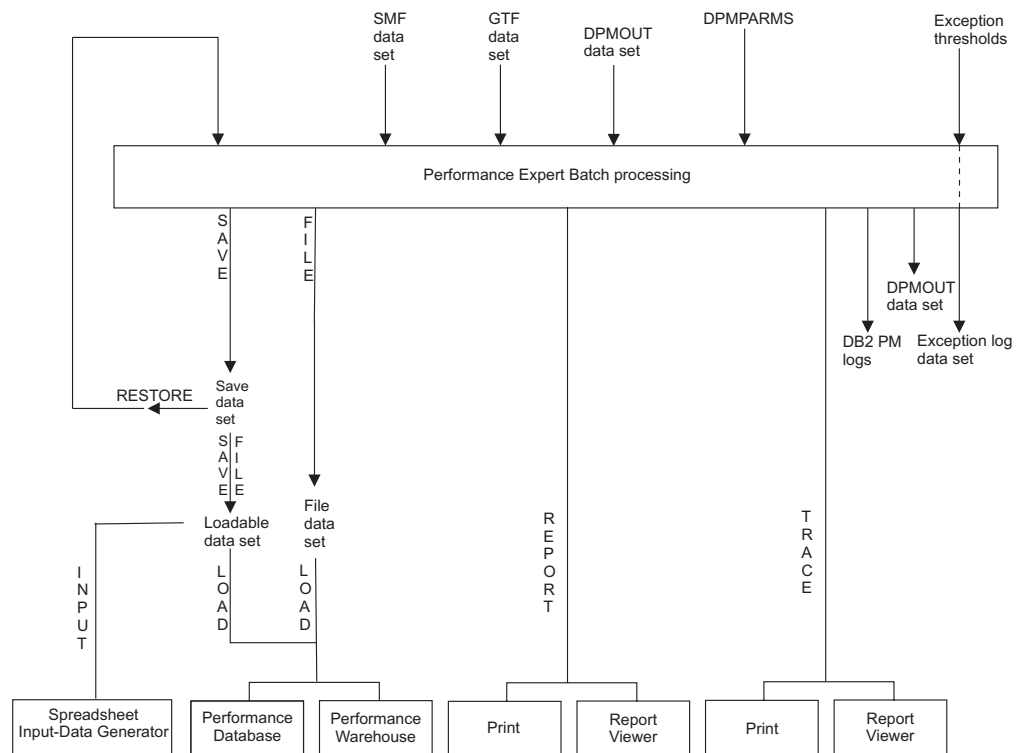


Figure 2. OMEGAMON XE for DB2 PE reporting

For information about input data sets, read Chapter 3, “Collecting data for reports,” on page 9, and for information about DPMPARMS, read Chapter 23, “Customizing OMEGAMON XE for DB2 PE functions,” on page 235. Exception thresholds are described in Chapter 8, “Using Statistics and Accounting reports to identify exceptions,” on page 65.

Chapter 3. Collecting data for reports

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

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

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

Note: OMEGAMON XE for DB2 PE provides sophisticated and flexible methods to collect performance data and to create reports from collected data. For both, the collection of data and the creation of reports, you have several choices to determine which data to collect and which data to include in reports. The basic rule is that your final reports will only show data that has previously been collected and has not been filtered out by any of the command parameters. Empty reports, processing counts of zero, or messages showing zero processed records usually indicate that required data within specified or required time intervals is not available in your input data or has been filtered by inappropriate command parameters during report generation. If required, refer to “Troubleshooting empty reports” on page 296 for more information.

“Managing DB2 traces with the Workstation Online Monitor” on page 10

The easiest way to produce input data for reports is to use the Workstation Online Monitor. You must specify the report set for which you want to collect data and the data set where it is to be collected. Then, you can use this data set as input for reports.

“Collecting report data with the ISPF Online Monitor” on page 10

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

“Collecting report data by specifying DB2 startup parameters” on page 20

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

“Collecting report data with the DB2 START TRACE command” on page 20

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

“Collecting report data with the Collect Report Data Batch program” on page 27

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

“Collecting report data with the Near-Term History Data Collector” on page 30
If the Near-Term History Data Collector is configured to store collected data in sequential data sets, the contents of these data sets can be used as input for OMEGAMON XE for DB2 PE reports and trace reports.

Managing DB2 traces with the Workstation Online Monitor

The easiest way to produce input data for reports is to use the Workstation Online Monitor. You must specify the report set for which you want to collect data and the data set where it is to be collected. Then, you can use this data set as input for reports.

You can start the collection of report data manually, or you can set it up to start automatically based on certain conditions, such as:

- A specified point in time
- When an exception threshold is reached
- an exception event is encountered

You can configure and start collect tasks for:

- One or more DB2 PM report sets
- Specific types of reports or traces within a report set
- Specific IFCIDs

You can also limit the data to specific:

- Locations
- Plan names
- Authorization IDs

You can configure DB2 traces to start:

- At a specified time of the day
- When a specified periodic exception is detected
- When a specified exception event occurs
- Immediately

You can stop DB2 traces manually or configure traces to stop:

- After a specified elapsed time
- After a specified number of trace records have been collected
- After a thread has been terminated or reused
- After a particular IFCID has been collected a specified number of times

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

Collecting report data with the ISPF Online Monitor

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

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

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

You can configure and start up to four independent collect tasks. With each task you can collect trace data for:

- One or more reports sets
- Specific types of reports or traces within a report set
- Specific IFCIDs

In addition, you can limit the data to specific:

- Locations
- Plan names
- Authorization IDs

You can configure DB2 traces to start:

- At a specified time of the day
- When a specified periodic exception is detected
- When a specified exception event occurs
- Immediately

You can stop DB2 traces manually or configure traces to stop:

- After a specified elapsed time
- After a specified number of trace records have been collected
- After a thread has been terminated or reused
- After a particular IFCID has been collected a specified number of times

“Accessing the Collect Report Data (CRD) panels”

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

“Configuring a collect task” on page 13

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

“Starting and stopping traces” on page 17

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

“Displaying trace status and messages” on page 19

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

“DB2 trace termination” on page 20

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

Accessing the Collect Report Data (CRD) panels

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

About this task

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

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

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

The Collect Report Data panel is displayed:

DGOMAP00 Collect Report Data

PM01DL0C DSN1 V10

For any trace task enter one of the following actions:

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

Task Description	Status
Collect data for acct/stats/audit	Not yet started
Test case for buffer overruns	Collecting data
Collect Task C	Never configured
Collect Task D	Never configured

Command ==> _____

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

Figure 3. Collect Report Data panel

From the Collect Report Data panel you can configure and control the collection of report data.

- Option 1 (Configure) displays a window where you configure collect tasks to collect report data and where you can limit the collection of DB2 trace data to time periods or events of interest. Refer to “Configuring a collect task” on page 13 for more details.
- Option 2 (Start) displays a window where you start a collect task after it has been configured. Starting a collect task actually starts a DB2 trace and enables the start and stop criteria that were specified with option 1 (Configure). Refer to “Starting and stopping traces” on page 17 for more details.
- Option 3 (Display) displays a window that shows the status of a collect task and any messages issued by that collect task. Refer to “Displaying trace status and messages” on page 19 for more details.
- Option 4 (Stop) displays a window where you can stop a collect task and associated DB2 traces. Stopping a collect task prevents the triggering of DB2 traces and stops all traces that were previously started by the task. Refer to “Starting and stopping traces” on page 17 for more details.

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

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 in the Collect Report Data panel and press Enter. The Trace Configuration window is displayed:

```

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 overtyp e 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 overtyp e with a blank to delete the selection.

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

_1024 OP Buffer size (64 - 65536 kilobytes)
__80  OP Buffer trigger (10 - 90 percent)

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

```

Use this window to specify the trace trigger method, the report sets, and the types of data to be collected.

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

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

Use the fields on the bottom part of this panel to specify whether to restrict the collection of data to specific data types (see “Restricting data types for selected report sets”), IFCIDs (see “Restricting IFCIDs for selected report sets” on page 15), or OMEGAMON XE for DB2 PE identifiers (see “Restricting identifiers for selected report sets” on page 16). 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 OP_n buffer that is used for collecting the data. The valid range is from 64 to 65536.

Use the OP Buffer trigger field to specify the percentage of the filled OP_n buffer when the **Collect task** is posted by DB2 to read the OP_n buffer. The valid range is from 10 to 90.

“Restricting data types for selected report sets”

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

“Restricting IFCIDs for selected report sets” on page 15

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

“Restricting identifiers for selected report sets” on page 16

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

Restricting data types for selected report sets

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

About this task

If you selected the Data Type field on the Trace Configuration window, the Data to Collect window (Figure 5 on page 15) is displayed once for each selected report set that has more than one data type.

The Data to Collect window shown in Figure 5 on page 15 uses an example for the Audit report set.

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 overwrite 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 5. Data to Collect Window

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

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

Restricting IFCIDs for selected report sets

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

About this task

If you selected the IFCID field on the Trace Configuration window, the IFCID Selection window (Figure 6 on page 16) is displayed.

Note: The IFCID Selection 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 6. IFCID Selection window

You can use the Select/Deselect all field to select or deselect all fields in this panel.

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

Restricting identifiers for selected report sets

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

About this task

If you selected the Requesting Location, Plan name and Authid field in the Trace Configuration window, the Trace Qualification window (Figure 7 on page 17) is displayed. Only data from threads that match the trace qualification criteria in this window will be collected. These trace qualification criteria are also used if thread termination is indicated in the Trigger by Time window (Figure 8 on page 18).

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 7. Trace Qualification window

Specify the name of the requesting location, plan name, and authorization ID, then press Enter. Do not specify multiple entries for more than one identifier, otherwise the number of DB2 traces started by the collect task could exceed the DB2 limit of 32 traces.

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

Starting and stopping traces

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

About this task

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

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

The Trigger by Time window (Figure 8 on page 18) is displayed if you specified that the trace is triggered by time. Use this window to specify a particular output data set name, and to set the start and stop trigger criteria for the collect task. You can set the DB2 traces to start at a specified time, and to stop after a specified number of minutes have passed, a specified number of records or IFCIDs have been collected, or a thread matching the trace qualification criteria has terminated.

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 8. Trigger by Time window

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

RECFM:

VBS

LRECL:

32756

BLKSIZE:

6233

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

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

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

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

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

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

- In the Trigger Immediately window, there are no start criteria because the DB2 traces are started immediately when the respective collect task is started in the Collect Report Data panel.

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

Displaying trace status and messages

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

About this task

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

```

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 . : 05/22/13 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 9. 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 about the status of the collect task.

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

The Active Traces for this Destination section of this window lists all active DB2 traces started by the task, and shows the trace type, trace class, output buffer destination, and qualification criteria for each active DB2 trace started by the task.

If many DB2 traces are listed, use the scrolling keys F7 (Up) and F8 (Down) to browse the list.

DB2 trace termination

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

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

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

Collecting report data by specifying DB2 startup parameters

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

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

Collecting report data with the DB2 START TRACE command

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

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

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

The following diagram shows the command syntax.

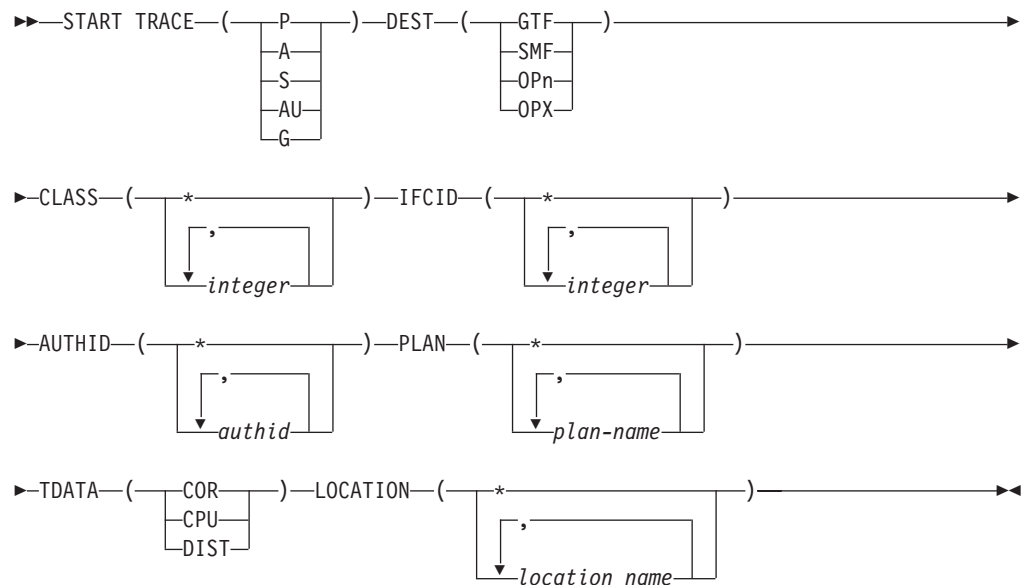


Figure 10. DB2 START TRACE command syntax

The **START TRACE** command accepts the following parameters and options:

TRACE

Specifies the type of data to be traced.

- P** Performance data
- A** Accounting data
- S** Statistics data
- AU** Audit data
- G** Global data

DEST Specifies the destination to which the traced data is directed. The destination can be the SMF (System Management Facility), the GTF (Generalized Trace Facility), or an OP buffer.

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

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

If you specify SMF as destination:

1. Ensure that SMF is active before you start the trace. For more information, see *z/OS MVS System Management Facilities (SMF)*.
2. Ensure that SMF collects the following records:
 - DB2 Accounting records (SMF type 101 records)
 - DB2 Audit records (SMF type 102 records)
 - DB2 Statistics records (SMF type 100 records)
 - DB2 Performance records (SMF type 102 records)
3. Ensure that the SMF data sets and buffers are large enough to hold the data being collected. If the SMF buffers run out of space, SMF rejects any records while the shortage of space exists. Lost data cannot be

recaptured. However, a long Statistics report or trace shows the number of records that have been lost in the DB2 IFC Destination block of the report.

CLASS

Specifies one or more trace classes within the trace type.

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

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

Note: Use this option to start IFCID 381, IFCID 497, or IFCID 498. You cannot start these IFCIDs using a DB2 performance class.

AUTHID and PLAN

Specifies authorization IDs and plan names to be traced.

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

TDATA

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

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

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

Table 2. Product section headers relevant to OMEGAMON XE for DB2 PE

Report set	COR	CPU	DIST
Accounting	Required	Not used	Required if distributed data is present
Audit	Required	Not used	Required if distributed data is present
I/O Activity	Required	Not used	Required if distributed data is present
Locking	Required	Not used	Required if distributed data is present

Table 2. Product section headers relevant to OMEGAMON XE for DB2 PE (continued)

Report set	COR	CPU	DIST
Record Trace	Used if present	Used if present	Required if distributed data is present
SQL Activity	Highly recommended	Recommended	Required if distributed data is present
Statistics	Not used	Not used	Required if distributed data is present
Utility Activity	Required	Recommended	Required if distributed data is present

LOCATION

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

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

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

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

LOCATION has no effect when Accounting class 2 is started.

“START TRACE command examples”

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

“DB2 instrumentation data” on page 24

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

START TRACE command examples

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

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

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

To collect Audit class 2 data, enter:

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

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

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

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

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

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

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

```
-START TRACE(A) CLASS(*)  
-START TRACE(A) CLASS(1,2,3,5,7,8)
```

The default destination for Accounting, Statistics, and Audit trace types is SMF, but you can route the trace data to GTF by specifying DEST(GTF) or to both SMF and GTF by specifying DEST(SMF,GTF) in the **START TRACE** command.

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

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

DB2 instrumentation data

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

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

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

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

- *Statistics* data shows to what extent the DB2 system services and database services are used. You can use this information to plan DB2 capacity and to tune an entire set of DB2 programs. Statistics data also contains information about deadlocks, timeouts, and DDF exception events.

The Statistics trace is written at specified intervals. You can control the Statistics interval by using STATISTICS TIME in the DSNTIPB installation panel.

- *Accounting* data provides information related to application programs and packages.

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

- *Performance* data provides information for performance analysis, performance tuning, and includes records of specific events in the system.
When you start a Performance trace, specify what you want to report, for example, I/O only or SQL only.
- *Audit* data provides information about DB2 security controls. You can use the data to ensure that data access is allowed only for authorized purposes.
Tables are the only auditable objects. To audit a table, include the AUDIT clause in the CREATE TABLE or ALTER TABLE statement.

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

The following table lists trace types, trace classes, and IFCIDs relevant to reports. Use the table to see which DB2 trace data is used as input for each report set.

Table 3. Input for OMEGAMON XE for DB2 PE report sets

Report set	DB2 trace type	DB2 trace class	Description of DB2 trace class	DB2 IFCIDs used
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
		10	Package Details	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, 319
		8	Utility access to any object	24, 105, 107
		10	Established trusted connections	269, 270
I/O Activity	Performance	4	Buffer manager I/O and EDM pool requests	6, 7, 8, 9, 10, 29, 30, 107
		5	Log manager	34, 35, 36, 37, 38, 39, 40, 41, 114, 115, 116, 119, 120
		21	Data sharing	107, 255

Table 3. Input for OMEGAMON XE for DB2 PE report sets (continued)

Report set	DB2 trace type	DB2 trace class	Description of DB2 trace class	DB2 IFCIDs used
Locking	Statistics	3	Deadlock and timeout information	172, 196
	Performance	4	Buffer manager I/O and EDM pool requests	226, 227
		6	Locking information	20, 44, 45, 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 types, 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
SQL Activity	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 trace classes	188, 324
Statistics	Statistics	1	Statistics data	1, 2
		6	Storage manager pool summary statistics	225
		8	Buffer pool data set statistics	199
		9	Aggregated accounting statistics	369

Table 3. Input for OMEGAMON XE for DB2 PE report sets (continued)

Report set	DB2 trace type	DB2 trace class	Description of DB2 trace class	DB2 IFCIDs used
System parameters	Performance	Any	These IFCIDs are available in all trace classes	106, 201, 202, 256
	Statistics	5	Data sharing global information	230
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, 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

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

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

Collecting report data with the Collect Report Data Batch program

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

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

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

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

The commands and parameters that determine the collection and postprocessing of data must be specified in the SYSIN DD statement of the respective JCL, as outlined in the following syntax diagram. A sample batch job can be found in data set *prefix.TK02SAMP(FPEZCRDJ)*.

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

VOLUME

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

UNIT

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

DSSIZE

Is optional. It can be used to override the default space unit that is used for the allocation of each GDG member. Possible values for *dsSizeType* are:

TYPE1

Causes the GDG member to be allocated with:

Space unit = BLK
Primary quantity = 100
Secondary quantity = 500

These values are the default.

TYPE2

Causes the GDG member to be allocated with:

Space unit = TRK
Primary quantity = 100
Secondary quantity = 500

TYPE3

Causes the GDG member to be allocated with:

Space unit = TRK
Primary quantity = 100
Secondary quantity = 1500

TYPE4

Causes the GDG member to be allocated with:

Space unit = CYL
Primary quantity = 10
Secondary quantity = 150

TYPE5

Causes the GDG member to be allocated with:

Space unit = CYL
Primary quantity = 10
Secondary quantity = 150

OPBUFSIZE

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

2048

Is the default size.

opSize

Can be up to 16384 KB.

ELAPSEDTIME

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

60 Is the default duration (in seconds).

secs
Specifies the duration (in seconds).

RECCOLLECTED

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

recs
Specifies the maximum number of IFCID records to collect.

Not supported in combination with GDGDATASET.

TERMTHREADS

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

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

Not supported in combination with GDGDATASET.

TRACE

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

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

```
:  
TRACE('START TRACE(PERFM) CLASS(30) IFCID(314) TDATA(CORRELATION,CPU)')  
TRACE('START TRACE(AUDIT) CLASS(31) IFCID(140,83)')
```

Collecting report data with the Near-Term History Data Collector

If the Near-Term History Data Collector is configured to store collected data in sequential data sets, the contents of these data sets can be used as input for OMEGAMON XE for DB2 PE reports and trace reports.

The Near-Term History Data Collector can be configured to store data in various types of data sets, for example sequential data sets, Generation Data Group (GDG) data sets, and VSAM data sets. The configuration is done with the Configuration Tool, the currently active configuration can be viewed by means of the OMEGAMON XE for DB2 PE Classic Interface in the Near-Term History Data Collection Options panel.

Ensure that collected data is written to sequential data sets by one of the following methods:

- The near-term history data collection options keyword **WRITEOPTION** in data set member **RKD2PAR(COPT<ssid>)** should have the value **VSAM,SEQ**.
- In the Near-Term History Data Collection Options panel you should see **Writeoption=VSAM,SEQ**. This panel also lists the currently used data sets where collected data is stored.

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

Chapter 4. Creating reports using commands

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

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

- Setup information
- General filters and controls
- Commands
- Subcommands and subcommand options

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

“Setup information for data sets”

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

“General filters and controls” on page 34

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

“Report set commands” on page 34

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

“Subcommands and subcommand options” on page 35

You can use subcommands and subcommand options to specify how you want the data to be presented.

“Example of a command string” on page 36

This example shows a JCL that produces a short Accounting report and a long Statistics trace.

Setup information for data sets

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

“ddnames of general data sets”

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

“ddnames of report data sets” on page 33

Output from OMEGAMON XE for DB2 PE report set processing is written to the report data sets. You must specify a data set for the SAVE, RESTORE, or FILE output corresponding to the particular report set you are requesting.

ddnames of general data sets

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

The ddnames for the required data sets are:

INPUTDD

The data set that contains the DB2 trace data to be used as input for OMEGAMON XE for DB2 PE reports. You must specify this fully qualified data set name.

STEPLIB

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

SYSIN

The data set that contains OMEGAMON XE for DB2 PE command string. If the command string is contained in the JCL, specify an asterisk (*), otherwise specify the fully qualified data set name.

The ddnames for the optional data sets are:

DPMLOG

The data set where processing messages are written.

DPMOUTDD

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

DMPPARMS

The data set that contains information about changes that you have made to OMEGAMON XE for DB2 PE standard processing settings. The things you can tailor are:

- Report layouts
- Time zone specifications (member LOCDATA)
- Correlation translation information (member CORRDATA)
- Exception field descriptions (member EXCHANGE)
- Definition of the main packages used in reporting (MAINPACK)

For more information, see Chapter 23, “Customizing OMEGAMON XE for DB2 PE functions,” on page 235.

ERRDMPDD

This service data set provides potentially corrupted IFCID data identified by the Reporter. You should only specify this ddname if corrupted IFCID records are reported in your JOBSUMDD or DPMLOG.

JOBSUMDD

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

JSSRSDD

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

SYSOUT

The data set where messages about sorting are written.

The following three data sets are used for exception processing. For more information, see Chapter 8, “Using Statistics and Accounting reports to identify exceptions,” on page 65.

EXCPTDD

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

EXFILDD1

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

EXTRCDD1

The data set where the Exception Log data is written.

ddnames of report data sets

Output from OMEGAMON XE for DB2 PE report set processing is written to the report data sets. You must specify a data set for the SAVE, RESTORE, or FILE output corresponding to the particular report set you are requesting.

The default ddnames for these data sets all start with a two-letter prefix that indicates the report set. The following table shows the report set and the corresponding prefix:

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

Report set	Prefix
Accounting	AC
Audit	AU
I/O Activity	IO
Locking	LO
Record Trace	RT
SQL Activity	SQ
Statistics	ST
System Parameters	SY
Utility Activity	UT

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

***cc*RPTDD**

The data set where report output is written.

***cc*TRCDD1**

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 *cc*TRCDD2, the third to *cc*TRCDD3, the fourth to *cc*TRCDD4, and the fifth to *cc*TRCDD5.

***cc*FILDD1**

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

***cc*SAVDD**

The data set where data is stored using the SAVE subcommand.

***cc*RSTDD**

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

***cc*WORK**

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

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

EXPLAIN

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

General filters and controls

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

These commands are shared by the various report sets.

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

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

GLOBAL

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

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

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

GROUP

Use the GROUP command to define a group of OMEGAMON XE for DB2 PE identifier values for use in reporting. See “Group data” on page 231.

LIST Use the LIST command to define a list of OMEGAMON XE for DB2 PE identifier values for use in reporting. See “Use lists” on page 231.

Report set commands

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

The default subcommands usually produce the shortest report in the report set. See “Report sets” on page 4 for an overview of the different report sets. For details on how to specify commands refer to the *Report Command Reference*.

Table 5. Report sets and report set commands

Report set	Command
Statistics report set	STATISTICS
Accounting report set	ACCOUNTING
Explain report set	EXPLAIN

Table 5. Report sets and report set commands (continued)

Report set	Command
SQL Activity report set	SQLACTIVITY
System Parameter report set	SYSPARMS
Utility Activity report set	UTILITY
Locking report set	LOCKING
I/O Activity report set	IOACTIVITY
Record Trace report set	RECTRACE
Audit report set	AUDIT

Subcommands and subcommand options

You can use subcommands and subcommand options to specify how you want the data to be presented.

For details on how to specify commands refer to the *Report Command Reference*.

REPORT

Use this subcommand to generate reports. In reports, data is summarized by OMEGAMON XE for DB2 PE identifiers, such as the primary authorization ID or the plan name.

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

TRACE

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

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

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

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

REDUCE

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

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

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

SAVE You can save data sets as follows:

Without CONVERT option

Saves reduced data into a VSAM data set. You can use the saved data in later reporting.

You can also convert the data set into a sequential data set that can be loaded into DB2 tables using the Save-File utility.

With CONVERT option

Converts and saves reduced data into a sequential data set that can be loaded into DB2 tables.

RESTORE

Use this subcommand to include previously saved data from the VSAM data set.

Example of a command string

This example shows a JCL that produces a short Accounting report and a long Statistics trace.

The text that follows describes the contents of the JCL and shows an example output from this JCL.

```
EDIT ---- SYS92226.T092210.RA000.USERPMA.R000003 ----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 //USERPMA JOB (TTS1,YUS7),'ANDREW',
0002 //          MSGCLASS=V,CLASS=D,NOTIFY=USERPM
0003 //          EXEC PGM=FPECMAIN,PARM='DATEFORMAT=MM/DD/YY'
0004 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0005 //INPUTDD DD DSN=OMPE.RK02DATA(FPECIVPI),DISP=SHR
0006 //JOBSUMDD DD SYSOUT=A
0007 //SYSIN DD *
0008 GLOBAL
0009 FROM (04/01/13,08:00)
0010 TO (04/01/13,09:00)
0011 ACCOUNTING
0012 REPORT
0013 FROM (04/01/13,08:48)
0014 TO (04/01/13,08:52)
0015 STATISTICS
0016 TRACE
0017 LAYOUT(LONG)
0018 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 11. Specifying a job stream using ISPF/PDF editor

The first two lines are user and environment settings. Change these settings according to your operating environment.

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

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

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

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

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

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

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

You must include the EXEC statement (line 18) to generate the reports. Otherwise, no report is generated.

Because no explicit output data sets are specified in this example, the reports will be written to ACRPTDD for the Accounting report and STTRCDD1 for the Statistics trace, which are the default ddnames.

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

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

Figure 12. Submitting the job

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

Accounting report - model layout SHORT

The following example shows an Accounting report produced in the previous JCL.

LOCATION: OMPD951		OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)						PAGE: 1-1		
GROUP: N/P		ACCOUNTING REPORT - SHORT						REQUESTED FROM: 04/01/13 08:48:00.00		
MEMBER: N/P								TO: 04/01/13 08:52:00.00		
SUBSYSTEM: D951		ORDER: PRIMAUTH-PLANNAME						INTERVAL FROM: 04/01/13 08:49:47.77		
DB2 VERSION: V10		SCOPE: MEMBER						TO: 04/01/13 08:50:34.04		
		#OCCURS	#COMMIT	INSERTS	OPENS	PREPARE	CLASS2	EL.TIME	BUF.UPDT	LOCK SUS
PRIMAUTH		#DISTR	SELECTS	UPDATES	CLOSES	CLASS1	EL.TIME	CLASS2	CPUTIME	SYN.READ
PLANNAME		#ROLLBK	FETCHES	MERGES	DELETES	CLASS1	CPUTIME		GETPAGES	TOT.PREF

DNSK		1	1	0.00	0.00		0.00	0.051437	N/P	0.00
'BLANK'		0	0.00	0.00	0.00		0.051442	0.000388	N/P	0
		0	0.00	0.00	0.00		0.000394	N/P	N/P	

DNSK		1	1	0.00	1.00		1.00	0.027885	0.00	0.00
DSNREXX		0	0.00	0.00	1.00		0.030849	0.002257	0.00	0
		0	1.00	0.00	0.00		0.003204	19.00	0.00	

PROGRAM NAME		TYPE	#OCCURS	#ALLOCS	SQLSTMT	CL7	ELAP.TIME	CL7 CPU TIME	CL8 SUSP.TIME	CL8 SUSP

DSNREXX	PACKAGE	1	1	5.00	0.027877	0.002249	0.000000	0.00
---------	---------	---	---	------	----------	----------	----------	------

*** TOTAL ***

DN SK	2	2	0.00	0.50	0.50	0.039661	0.00	0.00
	0	0.00	0.00	0.50	0.041146	0.001322	0.00	0
	0	0.50	0.00	0.00	0.001799	19.00	0.00	

PROGRAM NAME	TYPE	#OCCURS	#ALLOCS	SQLSTMT	CL7 ELAP.TIME	CL7 CPU TIME	CL8 SUSP.TIME	CL8 SUSP
ALL PROG	PACKAGE	1	1	5.00	0.027877	0.002249	0.000000	0.00

ACCOUNTING REPORT COMPLETE

Chapter 5. Creating reports with the Interactive Report Facility

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

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

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

“Selecting reports and reports details”

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

“Specifying default options by using the OPTIONS command” on page 41

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

“Composing and executing the batch report command stream” on page 41

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

“Saving and recalling selections” on page 42

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

“Example of producing an Accounting report” on page 42

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

Selecting reports and reports details

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

About this task

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

Interactive Report Selections

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	-
System Parameters	-
Exception log	-
Explain	-

Command ==> _____

F1=Help	F2=Split	F3=Exit	F5=Compose	F6=Browse	F9=Swap
F10=Global	F11=Incl excl	F12=Cancel			

Figure 13. Interactive Report Selections panel

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

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

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

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

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

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

Specifying default options by using the **OPTIONS** command

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

About this task

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

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

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

For each ddname, enter one of the following to update the data set information:

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

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

Press Enter to validate the entries.

Composing and executing the batch report command stream

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

About this task

If foreground processing was selected, the job is executed immediately. Your terminal remains busy until the job is completed.

If background processing was selected, the Job Processing Selections panel is displayed. In this panel, you can:

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

Saving and recalling selections

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

About this task

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

RECFM

FB

LRECL

80

BLKSIZE

6160

Directory blocks

Depends on how many members you want to save.

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

Example of producing an Accounting report

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

About this task

Comprehensive help information is available on all IRF panels.

Procedure

1. Start the IRF by invoking option 1 (Create and execute reporting commands) from the main menu. The Interactive Report Selections panel is displayed:

```

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 . . . . . -
System Parameters . . . . . -
Exception log . . . . . -
Explain . . . . . -

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

```

Figure 14. Interactive Report Selections panel

2. Type RESET on the command line to delete all previously made selections from the panel.
3. Select an Accounting report by entering a forward slash (/) on the corresponding line, as shown in the previous figure.
4. Type OPTIONS on the command line. The Session Options window is displayed, overlaying the Interactive Report Selections panel:

```

Interactive Report Selections

S
R
A
S
S
L
I
A
U
R
A

      Session Options
      Update fields as required, then press Enter.
Confirmation display . . . . . 1 1=yes 2=no
Initial menu choice . . . . . 1-6 or blank
Execution mode . . . . . 1 1=Background
                        2=Foreground
                        3=Prompt
DMPARMS data set . . .

      F1=Help      F2=Split      F9=Swap      F12=Cancel

Global Processing . . . . . -
System Parameters . . . . . -
Exception log . . . . . -
Explain . . . . . -

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

```

Figure 15. Interactive Report Selections panel - Session Options window

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

6. Press Enter. The Accounting Report Selections panel is displayed:

```

Accounting REPORT Selections

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

  DDname      User Comment
/  _____ My Accounting report short_____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____
-  _____

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

```

Figure 16. Accounting Report Selections panel

7. Type a forward slash (/) in the action field and give the report an appropriate name in the User Comment field, as shown in the previous figure. You are not required to specify a ddname. The default is used.
8. Press Enter. The Accounting Report panel is displayed:

```

Accounting REPORT

Update fields as required, then press Enter.

User comment . . . . . My Accounting report short_____
DDname . . . . . ACRPTDD_

Scope . . . . . 1      1=member 2=group

Layout . . . . . short_  User-tailored Report Format

Exception . . . . . 2  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=Incl excl  F12=Cancel

```

Figure 17. Accounting Report panel

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

9. Press Enter to return to the Accounting Report Selections panel.
10. Press F3 (Exit) to return to the Interactive Report Selections panel. You have completed the specifications for the commands, subcommands, options and keywords that are required to generate the example Accounting report.
11. You can use F6 (Browse) to view the command stream. This displays the following panel:

```

BROWSE - HECK.PM05.SPFTEMP1.CNTL ----- Line 00000000 Col 001 080
***** Top of Data *****
ACCOUNTING
      REPORT                      /*My Accounting report short*/
      DDNAME(ACRPTDD)
      LAYOUT(SHORT)
      SCOPE(MEMBER)
EXEC
***** Bottom of Data *****

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

```

Figure 18. Accounting report command stream

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

```

                                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     'HECK.ACC.INPUT.SMF'      *
      EXCPTDD
      EXTRCDD1
      EXFILDD1
      ACRPTDD
      DPMLOG       SYSOUT=A
      DPMOUTDD
      DPMPARMS
      JOBSUMDD
      JSSRSDD
      SYSOUT       SYSOUT=A
      SYSUDUMP
***** Bottom of data *****

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

```

Figure 19. Background DDname Selections panel

13. Enter at least the required information, marked by asterisks (*). Usually, you must specify the input data set that contains the data to be reported. In this

example, ACC.INPUT.SMF is entered in the INPUTDD line, which resolves to 'HECK.ACC.INPUT.SMF' (the TSO prefix is added) when you press Enter.

The list of ddnames also shows the default ddname for Accounting reports (ACRPTDD). Output data sets are dynamically allocated, if not specified explicitly. If you want to become familiar with the IRF and do not have real input data available, you can use the sample data set DB2PM.V400.DPMIN40. If this data set is not available, press F1 (Help) and F5 (Exhelp) for current information about the sample data set.

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

Job Processing Selections

Update the job statements as required, then select one of the following.

1. Browse the generated job stream
2. Edit the generated job stream
3. Store the job stream for future use
4. Submit the job stream for background execution

Job statement information

Command ===>

F1=Help F2=Split F3=Exit F9=Swap F12=Cancel

Figure 20. Job Processing Selections panel

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

```

BROWSE - HECK.PM05.SPFTEMP1.CNTL ----- Line 00000000 Col 001 080
Command ===> _____ Scroll ===> PAGE
***** Top of Data *****
//PMV110 EXEC PGM=DB2PM
//STEPLIB DD DSN=OMPE410.RKANMOD,DISP=SHR
//INPUTDD DD DSN=ACC.INPUT.SMF,DISP=SHR
//ACRPTDD DD SYSOUT=A
//DPMLOG DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SYSIN DD *
ACCOUNTING
          REPORT                      /*Account Report short*/
                                DDNAME(ACRPTDD)
                                LAYOUT(SHORT)
                                NOEXCEPTION
                                SCOPE(MEMBER)
EXEC
***** Bottom of Data *****

F1=Help    F2=Split    F3=Exit    F5=Rfind    F7=Up       F8=Down    F9=Swap
F10=Left   F11=Right   F12=Cancel

```

Figure 21. Interactive Report Facility (IRF) - example of generated job stream

16. To submit the job, specify appropriate job statement information at the bottom half of the Job Processing Selections panel and press Enter. You are prompted if this information is missing.

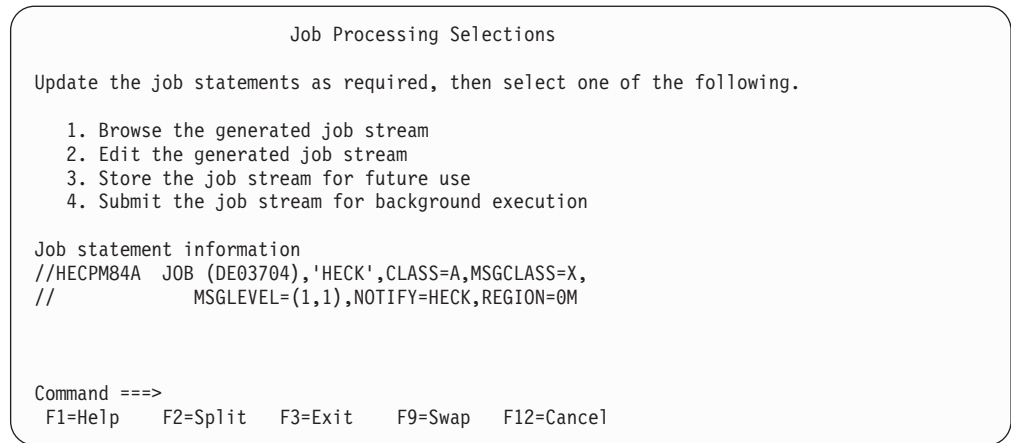


Figure 22. Job Processing Selections panel - job statement information

You should now see a message that indicates whether the job was successfully submitted.

Chapter 6. Using Statistics traces to get system overview information

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

To do this:

- Run a DB2 trace to collect DB2 Statistics data during a 24-hour period.
- Create a Statistics trace report.
- Interpret the report and adjust the system.

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

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

“Collecting Statistics data with a DB2 trace”

Use the Performance Expert Client to configure and start a DB2 trace to collect DB2 Statistics data during a 24-hour period.

“Creating a Statistics trace report” on page 50

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

“Interpreting the Statistics trace report and applying rules of thumb” on page 50

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

Collecting Statistics data with a DB2 trace

Use the Performance Expert Client to configure and start a DB2 trace to collect DB2 Statistics data during a 24-hour period.

About this task

To do this, start the Performance Expert Client:

1. On the menu bar of the **System Overview** window, click **Tools > Performance Warehouse - Expert** to open the **Performance Warehouse** window.
2. On the Performance Warehouse window, select the **Trace on z/OS** tab.
3. Open the folders to connect to the **z/OS** subsystem.
4. Switch to your private group. If not available, right-click the **Process Groups** node to create your private copy.
5. Expand the public tree
6. Copy the CRD (collect report data) process template from the **Public** process groups to a process group of your own.
7. Expand your private group tree.
8. Double-click the CRD step to open the **CRD Step Properties** window.
9. Open the **Options** page and change the following option categories:
 - For **Output data set**, type an output data set name and select **Append** or **Overwrite**.

- For **Data**, clear the **Statistics** check box. Specify which IFCIDs you want to collect. The IFCIDs associated with this trace are displayed and highlighted. Only the highlighted IFCIDs are collected in the trace. Here you can, for example, choose not to include buffer pool statistics at data set level.
 - For **Stop**, specify the criteria for the data collection. Select the **Elapsed time** check box and specify a time of 24:00:00.
 - Click **OK** to save the configuration of the CRD step.
10. When finished, right-click the CRD process and select **Execute** to open the **Performance Warehouse Process Scheduler** window to define a schedule for the selected process.
 11. Select the **Execute once, immediately** to run the process once.
 12. Click **Finish** to start the trace.

Creating a Statistics trace report

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

About this task

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

```

EDIT ---- ANDREW.OMPE.JOBS.STTRALON----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce a Statistics LONG trace report */
0002 //AWOMPES JOB (TTS1,YUS7),'ANDREW',
0003 //          MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 //          EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE1,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.STTRCA01,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 STATISTICS
0011     TRACE
0012     LAYOUT(LONG)
0013     FROM (07/16/13,00:01)
0014     TO (07/17/13,00:01)
0015 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 23. Specifying a job stream using ISPF/PDF editor

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

Interpreting the Statistics trace report and applying rules of thumb

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

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

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

“Getting an overview of your DB2 system performance”

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

“Buffer pool efficiency” on page 52

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

“EDM pool failures” on page 53

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

“Authorization problems” on page 53

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

“RID List failures” on page 53

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

“Locking problems” on page 54

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

“Logging performance” on page 54

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

“Data set Open and Close activities” on page 54

This report block of the Statistics trace report indicates whether enough data sets are allowed to be concurrently open.

“ROWID efficiency” on page 54

This report block shows how efficient direct row access is used. Direct row access is very fast because DB2 does not use an index or table space scan to find the row.

“Thread management” on page 54

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

Getting an overview of your DB2 system performance

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

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

- **DBM1 storage** DBM1 storage is allocated below and above the 2 GB bar and is used for buffer pools, EDM pools, RID pools, and authorization caches. It is essential that the various pools and caches are optimized in their allocated space.
- **Data set Open and Close activity** Applications that leave data sets open can affect other applications that must wait because too many data sets are already open.
- **Logging**
Logging activities interrupt normal DB2 operations. Logging problems will therefore affect applications and performance.
- **Locking**

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

- **ROWID**

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

- **Thread management**

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

Buffer pool efficiency

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

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

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

Hit ratios

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

The buffer pool hit ratio (BP00L HIT RATIO (%)) shows the number of Getpage requests issued by applications and satisfied by the buffer pool, expressed as a percentage of all Getpage requests. This is a relative value that depends on the type of application. For example, applications that browse large amounts of noncontiguous data could cause the buffer pool hit ratio to drop near 0.

Avoid page-Ins

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

A buffer pool that is too large can also cause performance problems because it causes paging to a hard disk drive.

Look at the PAGE-INS REQUIRED FOR READ and PAGE-INS REQUIRED FOR WRITE fields in the Buffer Pool Read block and the Buffer Pool Write block of the report. These values should be zero, or close to zero. Higher values mean that the buffer pool size is over allocated. Consider reducing the buffer pool size.

Optimize prefetch

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

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

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

- Review the Sequential Steal thresholds (VPSEQT)
These thresholds are percentages of the virtual buffer that might be occupied by sequentially accessed pages. If these thresholds are too low, prefetch can be disabled. If these thresholds are too high, other thresholds can be reached too soon.
- Reduce the Deferred Write thresholds (VDWQT and DWQT)
If the DB2 system is used mainly for high-use query systems reliant on prefetch, reducing these thresholds decreases the portion of the virtual buffer pool that is occupied by updated pages.
- Increase the size of the buffer pool
Use the **ALTER BUFFERPOOL** command to increase the size of the buffer pool, which will give the system more space for prefetched pages.

EDM pool failures

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

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

Authorization problems

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

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

If the number of package authorization failures (AUTH UNSUCC-PKG-CACHE) is not zero, or close to zero, use the ZPARM ZPAC to increase the size of the package cache.

RID List failures

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

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

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

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

Locking problems

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

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

Logging performance

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

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

Data set Open and Close activities

This report block of the Statistics trace report indicates whether enough data sets are allowed to be concurrently open.

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

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

ROWID efficiency

This report block shows how efficient direct row access is used. Direct row access is very fast because DB2 does not use an index or table space scan to find the row.

The TABLE SPACE SCAN USED field value should be zero. Use the WHERE clause to cause an unsuccessful direct row access to revert to a matching index scan using the primary key:

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

Thread management

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

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

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

The combined maximum allowed for MAX USERS and MAX REMOTE ACTIVE depends on the DB2 version installed. It cannot exceed:

20000 For DB2 10 or later.

2000 Prior to DB2 10.

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

Chapter 7. Using an Accounting report to analyze resource activities

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

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

A good first indicator for applications is the time used per thread by any particular user. A strategy to improving application performance is to identify the users with the most system time consumption and examine their most frequently used applications:

- Create a DB2 trace for input to a report
- Create a short Accounting report that identifies the users with the highest system time consumption.
- Interpret the report and identify users and plans that need attention
- Create an Accounting trace report for each problem user and plan
- Interpret the trace report and find remedial actions
- Apply rules of thumb in the Performance Warehouse
 - “Collecting Accounting data with a DB2 trace”
Use the Performance Expert Client to configure and start a DB2 trace to collect DB2 Accounting data during a 24-hour period.
 - “Creating a short Accounting report” on page 58
Use your ISPF editor to create a JCL job that creates a short Accounting report.
 - “Interpreting the Accounting report and making changes” on page 58
An example of a short Accounting report is evaluated and key fields are emphasized.
 - “Interpreting the Accounting trace report and applying rules of thumb” on page 60
Using an Accounting trace report for a specific problem analysis and applying expert rules of thumb.

Collecting Accounting data with a DB2 trace

Use the Performance Expert Client to configure and start a DB2 trace to collect DB2 Accounting data during a 24-hour period.

About this task

To do this, start the Performance Expert Client:

1. On the menu bar of the **System Overview** window, click **Tools > Performance Warehouse - Expert** to open the **Performance Warehouse** window.
2. On the Performance Warehouse window, select the **Trace on z/OS** tab.
3. Open the folders to connect to the **z/OS** subsystem.
4. Switch to your private group. If not available, right-click the **Process Groups** node to create your private copy.

5. Expand the public tree
6. Copy the CRD (collect report data) process template from the **Public** process groups to a process group of your own.
7. Expand your private group tree.
8. Double-click the CRD step to open the **CRD Step Properties** window.
9. Open the **Options** page and change the following option categories:
 - For **Output data set**, type an output data set name and select **Append** or **Overwrite**.
 - For **Data**, clear the **Accounting** check box. Specify which IFCIDs you want to collect. The IFCIDs associated with this trace are displayed and highlighted. Only the highlighted IFCIDs are collected in the trace. Here you can, for example, choose not to include buffer pool accounting at data set level.
 - For **Stop**, specify the criteria for the data collection. Select the **Elapsed time** check box and specify a time of 24:00:00.
 - Click **OK** to save the configuration of the CRD step.
10. When finished, right-click the CRD process and select **Execute** to open the **Performance Warehouse Process Scheduler** window to define a schedule for the selected process.
11. Select the **Execute once, immediately** to run the process once.
12. Click **Finish** to start the trace.

Creating a short Accounting report

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

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

```

EDIT ---- ANDREW.OMPE.JOBS.ACREPSHO----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce an accounting short report */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 //          MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 //          EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE2,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.ACRPT01,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 ACCOUNTING
0011 REPORT
0012 TOP(20)
0013 EXEC
***** ***** BOTTOM OF DATA *****
F1=Help      F2=Split    F3=Exit      F5=Rfind     F6=Rchange   F7=Up
F8=Down      F9=Swap      F10=Left    F11=Right    F12=Cancel

```

Figure 24. Specifying a job stream using ISPF/PDF editor

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

Interpreting the Accounting report and making changes

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

Short Accounting report

The output from your short Accounting report could look similar to the following report:

```
LOCATION: STM4D61Y          OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)          PAGE: 1-1
GROUP: N/P                ACCOUNTING REPORT - SHORT                      REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P                                                       TO: NOT SPECIFIED
SUBSYSTEM: Y71Y          INTERVAL FROM: 02/14/13 10:20:09.30
DB2 VERSION: V11         TO: 02/14/13 10:20:09.30

ELAPSED TIME SPENT IN APPLICATION          TOP NUMBER REQUESTED:    20

PRMAUTH                                #OCCURS #ROLLBK SELECTS INSERTS UPDATES DELETES CLASS1 EL.TIME CLASS2 EL.TIME GETPAGES SYN.READ LOCK SUS
PLANNAME                                #DISTRS #COMMIT  FETCHES  OPENS  CLOSSES PREPARE CLASS1 CPUTIME CLASS2 CPUTIME BUF.UPDT TOT.PREF #LOCKOUT
-----
JUSTIN                                6        0    0.00    0.00    0.00    0.00    2:01.610942    1:05.002506 10553.33  272.00    6.83
JUSDOIT3                             0       12    2.00    1.00    1.00    1.00    1:06.879690    39.838561  2585.33  140.80     0
JUSTIN                                6        1    0.00    0.00    0.00    0.00    1:51.610942    2:02.005506  993.53   84.33    3.51
JUSDOIT1                             0        8    3.00    1.00    1.00    1.00    56.879690     56.838591  3985.37  240.50     0
JUSTIN                                6        0    0.00    0.00    0.00    0.00    1:24.096142    42.125542 12285.31  84.33    8.63
JUSDOIT2                             0       12    2.00    1.00    1.00    1.00    45.687990     38.746239  3958.38  240.50     0
ANDREW                                1        0    0.00    0.00    0.00    0.00    1:10.506675    1:09.849935 11822.00  84.33    3.00
AWSPLAN3                             0        2    2.00    1.00    1.00    1.00    1:03.779540     1:03.737837 3084.00  270.00     0
ANDREW                                6        0    0.00    0.00    0.00    0.00    1:01.610942    1:22.057438 10553.33  84.33    1.89
AWSPLAN1                             0       12    2.00    1.00    1.00    1.00    56.879690     56.838591  3864.34  240.50     0
ANDREW                                5        0    0.00    0.00    0.00    0.00    59.831795     59.233317 10299.60  46.80    0.40
AWSPLAN1                             0       10    2.00    1.00    1.00    1.00    55.499719     55.458742  2485.60  234.60     0
.....
.....
.....
.....
ACCOUNTING REPORT COMPLETE
```

In this example, the application with the highest elapsed time is JUSTIN-JUSDOIT3. There are a couple of points of interest for this user application that can give you an idea of what to look for in a more detailed report:

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

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

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

```

EDIT ---- ANDREW.OMPE.JOBS.ACTRALON----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce an accounting LONG trace report */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE2,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.ACTRC01,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 ACCOUNTING
0011 TRACE
0012 LAYOUT(LONG)
0013 INCLUDE(PRIMAUTH(JUSTIN) PLANNAME(JUSDOIT1 JUSTDOIT2 JUSTDOIT3))
001 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 25. Specifying a job stream using ISPF/PDF editor

Interpreting the Accounting trace report and applying rules of thumb

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

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

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

“The number of Getpage requests per SQL statement” on page 61

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

“Comparison of class 1 and class 2 times” on page 61

The ratio of class 1 and class 2 times indicate whether a potential problem lies in DB2.

“Comparison of class 2 elapsed and class 2 CPU times” on page 61

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

“Class 2 CPU time per SQL DML statement execution” on page 61

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

“Not accounted time” on page 62

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

“Class 3 suspensions” on page 62

Class 3 suspensions indicate potential problems with lock acquisitions, long wait times for synchronous I/O, and long service task switch times.

The number of Getpage requests per SQL statement

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

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

A high Getpage/SQL ratio could be caused by:

- The use of table scan. This causes each page in the table to be accessed and scanned, even though only one row is returned.
- The use of nonmatching index scan. When this happens, one Getpage request is made for each data page in the table and each index page in the associated index, even though only one row is returned.
- The use of an index with low cardinality. This can result in multiple page scans before a row is returned.

Comparison of class 1 and class 2 times

The ratio of class 1 and class 2 times indicate whether a potential problem lies in DB2.

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

If the class 2 elapsed time is less than half of the class 1 elapsed time, the problem is not with DB2. Possible causes include:

- OS/390
- CICS
- IMS
- Other programs

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

Comparison of class 2 elapsed and class 2 CPU times

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

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

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

Class 2 CPU time per SQL DML statement execution

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

The time taken for an SQL DML statement to complete depends on the speed of your processor. Typically, for a 100 MIPS CPU, a DML statement should complete

in less than one millisecond. Because some DML statements take more cycles to complete than others, the statements are weighted as follows:
`SELECT+INSERT+UPDATE+OPEN+(FETCH * 0.1)+(PREPARE * 5).`

If you find that your SQL statement time is averaging well over the one millisecond, you can use the Workstation Online Monitor to identify the problem statements and use the drill-down capability to view these at statement level. If you suspect an access path problem, you can use Data Studio or InfoSphere® Optim™ Query Workload Tuner to show the access path selected.

Not accounted time

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

This is time that is not recorded by DB2, either in class 2 time or in class 3 suspension time. This time should be below 25% of the total class 2 time. A higher percentage usually indicates an overloaded server, which could cause problems for online transactions. When this figure is consistently high, you should consider reducing the workload on the server or installing a bigger server.

Class 3 suspensions

Class 3 suspensions indicate potential problems with lock acquisitions, long wait times for synchronous I/O, and long service task switch times.

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

When evaluating class 3 suspensions, check the following information.

Commit interval

This is calculated as:

`COMMIT INTERVAL=CLASS 2 ELAPSED TIME / (COMMIT+ROLLBACKS)`

A value of 1 to 5 is acceptable. For example, if CLASS 2 ELAPSED TIME is 1 second and COMMIT is 1, the result is 1. Higher values indicate problems with lock acquisition, especially in batch applications.

Class 3 synchronous I/O time per I/O event

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

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

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

Service task switch

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

The most common contributors to service task suspensions are:

- Wait for commit processing for updates (UPDATE COMMIT)
- Wait for OPEN/CLOSE service task (including HSM recall)
- Wait for SYSLGRNG recording service task

- Wait for data set extend/delete/define service task (EXT/DEL/DEF)
- Wait for other service tasks (OTHER SERVICE)

The significant fields here are the times used by:

- UPDATE COMMIT

This is the average service time associated with a DB2 commit, abort, or deallocation. Normally this value should be below 12 milliseconds. If this value is significantly higher, verify the application or raise the interval between commits.

- OPEN/CLOSE

If the average time for an OPEN/CLOSE exceeds 150 milliseconds, DSMAX can be too low or too many data sets are open in the system. Adjust the DSMAX threshold. A figure of between 8000 and 10000 is a good working size in most cases.

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

Chapter 8. Using Statistics and Accounting reports to identify exceptions

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

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

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

“Output from exception reporting”

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

“Specifying exceptions using the Exception Threshold data set editor” on page 66

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

“How DB2 instrumentation data matches exception thresholds” on page 70

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

“Example of producing an Accounting exception report” on page 71

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

“Which exception fields and threshold values to choose” on page 73

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

Output from exception reporting

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

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

Reports and traces

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

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

File data sets and Log File data sets

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

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

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

Specifying exceptions using the Exception Threshold data set editor

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

About this task

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

The threshold values in the Exception Threshold data set can also be modified by means of the exception profiling method, which uses a sample of DB2 instrumentation data to calculate and set individual threshold values in the Exception Threshold data set. Refer to “Exception profiling” on page 210 for more details. For more details about the Exception Threshold data set and a sample data set that can be used to get started, refer to “Exception Threshold data set” on page 209.

The Exception Threshold data set editor is accessed indirectly from the Data Set Maintenance Menu (DGOPMENU, (Figure 26 on page 67). To display this menu, select option 4 (Maintain parameter data sets) from the Online Monitor Main Menu.

```

DGPOMENU                                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
'DG0710.THRESH'_____

DPMPARMS data set
'DG0710.DPMPARMS'_____

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

```

Figure 26. Selecting Maintain Exception Threshold data set

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

Press Enter. The Exception Threshold Category Selection panel is displayed, as shown in Figure 27.

```

DGOPXDS1                                Exception Threshold Category Selection
Command ==>_____

Select one or more categories, then press Enter. Overtyping 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 27. Exception Threshold Category Selection panel

This panel shows the categories of exception threshold fields from which you can choose. The name of the category indicates the area where the exception applies:

- ... per Plan ... or ... per Program ... exceptions are reported in Accounting
- ... per System ... or ... per Address Space... exceptions are reported in Statistics

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

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

As an example, the Elapsed, CPU and Waiting Times per Plan Execution field is selected in Figure 27 on page 67. When you press Enter, the Exception Threshold Field Selection panel is displayed, which shows all fields associated with this category, as shown in Figure 28.

DG0PXDS2
Exception Threshold Field Selection

Command ==> _____

Select one or more fields, then press Enter. Overtyping 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 switch suspensions (Class 3)
-- End of Items --	

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

Figure 28. Exception Threshold Field Selection panel

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

Select a field for which you want to specify the exception thresholds. Type a forward slash (/) or S in the selection field and press Enter. The Exception Threshold Field Details panel is displayed (Figure 29 on page 69).

- The thresholds. You can specify two kinds of thresholds for a field: Warning threshold and Problem threshold. Specify a warning threshold value to alert you to potential problems and a problem threshold value to indicate a more serious condition.
- The additional criteria for the data for which the exception thresholds apply. These criteria are useful if you want to specify different exception threshold values depending on the environment. For example, you might want to specify different Elapsed Time thresholds for online transactions and batch jobs, in which case you would supply a specific connection name in this panel, press F5 (Add), and specify a different connection name in the second panel. You can specify a generic name using an asterisk (*).

Usage Notes:

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

How DB2 instrumentation data matches exception thresholds

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

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

The key fields of a record are:

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

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

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

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

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

Example of producing an Accounting exception report

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

Accounting report - without Exception

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

LOCATION: DSNDB0G GROUP: DSNDB0G MEMBER: DB1G SUBSYSTEM: DB1G DB2 VERSION: V11		OMEGAMON XE for DB2 Performance Expert (V5R3M0) ACCOUNTING REPORT - SHORT ORDER: PRIMAUTH-PLANNAME SCOPE: MEMBER						PAGE: 1-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 05/15/13 22:35:07.26 TO: 05/15/13 22:56:45.89						
PRIMAUTH PLANNAME		#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK SUS #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
DSNTEP71		0	7	1.14	0.57	0.57	1.57	46.990679		46.971450	38572.71	1018.14		0
....														

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

To save time, you can specify exception thresholds for the class 1 time field per thread, and run Accounting exception reports that show only entries that have exceptionally high values for this field. You can further limit the data by only checking online transactions because you would expect to find a high elapsed time for batch jobs.

EXCEPTION
:

Accounting exception report

The following example shows the resulting Accounting exception report.

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

If you specify more than one exception threshold, all records that contained any fields that reached an exception threshold value will be listed.

```
LOCATION: DSNDB0G          OMEGAMON XE for DB2 Performance Expert (V5R3M0)          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: 05/15/13 22:35:07.26
DB2 VERSION: V11          SCOPE: MEMBER          TO: 05/15/13 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  CLOSSES 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  *
*               ADRECETT  ELAPSED TIME IN APPLICATION (CLASS 1)  TOTAL          5.069463 > 0  *
*               *               *               *               *               *
*****

LOCATION: DSNDB0G          OMEGAMON XE for DB2 Performance Expert (V5R3M0)          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: 05/15/13 22:47:05.09
DB2 VERSION: V10          SCOPE: MEMBER          TO: 05/15/13 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  CLOSSES 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  *
*               ADTDDL    TOTAL SQL DDL STATEMENTS  TOTAL          0 < 1  *
*               *               *               *               *               *
*****
:
```

Which exception fields and threshold values to choose

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

To get started with exception reporting, consider using the sample Exception Threshold data set member RK02DATA(DG0ETV51), which contains a selection of predefined exception thresholds. For more information, see “Exception Threshold data set” on page 209.

In general, long response times are a good indicator of a performance problem and therefore you should start by defining exception thresholds for time fields.

To use exception processing efficiently, consider what the most important applications or transactions in your system are. Always define exception thresholds for critical business applications. In addition, frequently executed applications are good candidates for exception thresholds.

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

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

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

You can use the exception profiling method and a sample of your installation's DB2 instrumentation data to calculate threshold values for exception fields. For more information, see "Exception profiling" on page 210.

Chapter 9. Using SQL Activity reports to monitor SQL performance

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

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

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

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

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

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

By default, no workload information is shown.

When you request workload, extra information is also provided:

Accounting

If IFCID 3 is included in your DB2 trace, OMEGAMON XE for DB2 PE includes an long Accounting trace for each thread.

Minibind

This shows bind information for each plan step. This is information taken from IFCID 22 and shows much of the information contained in the PLAN_TABLE.

User-defined functions

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

You can change the organization of data to group (summarize) events by:

- Cursor
- Program
- Statement number

- Statement type

Or you can change all of the above.

You can also sort the data by:

- Time:
 - Elapsed
 - Exit
 - I/O
 - Suspension
 - TCB
- Number of:
 - Exits
 - I/O requests
 - Merge passes for each sort
 - Pages scanned
 - Records sorted
 - Rows processed
 - Scans
 - Suspensions
 - Work files sorted

“When to use SQL Activity reports”

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

“Input for SQL Activity reports” on page 77

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

“Collecting SQL Activity data with a DB2 trace” on page 79

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

“Creating an SQL Activity trace” on page 79

Use your ISPF editor to create a JCL job that creates an SQL Activity trace.

“Interpreting the result” on page 82

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

“Remark about lengths of SQL statement texts in SQL Activity traces” on page 82

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

When to use SQL Activity reports

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

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

Input for SQL Activity reports

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

SQL activity events are classified as follows:

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

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

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

Table 6. IFCIDs used for SQL Activity reports

Start IFCID	DB2 trace class & SQL Activity event type	End IFCID
3 Accounting	• 1 Accounting data • 2 DB2 times • 3 Suspension, system events • 5 Package in DB2 time	None
6 Read I/O Start	4 I/O	7 Read I/O Stop
8 Write I/O Synch begin	4 I/O	9 Write I/O end
11 Validate Exit	13 Exit	
12 Edit Exit to Encode	13 Exit	
15 Index Scan Begin	8 Scan	18 Scan End
16 Insert Scan Begin	8 Scan	18 Scan End
17 Sequential Scan Begin	8 Scan	18 Scan End
19 Edit Exit to Decode	13 Exit	
20 Lock Summary	6 Locking	None
22 Minibind	3 Base	None
28 Sort Phase Detail	9 Sort	None
44 Lock Suspend	6 Locking	45 Lock Resume
53 SQL Describe, Commit, Rollback, or Remote statement	3 Base	None

Table 6. IFCIDs used for SQL Activity reports (continued)

Start IFCID	DB2 trace class & SQL Activity event type	End IFCID
55 Set SQLID	3 Base	None
59 Fetch Start	3 Base	58 End SQL
60 Select Start	3 Base	58 End SQL
61 Insert, Update, or Delete Start	3 Base	58 End SQL
62 DDL Start	3 Base	58 End SQL
63 SQL Statement	3 Base	None
64 Prepare Start	3 Base	58 End SQL
65 Open Cursor	3 Base	58 End SQL
66 Close Cursor	3 Base	58 End SQL
68 Rollback Start	3 Base	69 Rollback End
70 Commit Phase 2 Start	3 Base	71 Commit Phase 2 End
72 Create Thread Start	3 Base	73 Create Thread End
74 Terminate Thread Start	3 Base	75 Terminate Thread End
84 Prepare Start	3 Base	85 Prepare End
86 Signon Start	3 Base	87 Signon End
88 Synch Start	3 Base	89 Synch End
92 AMS Command Start	3 Workload highlight	97 AMS Command End
95 Sort Start	3 Sort	96 Sort End
105 DBID/OBID Translation	10 Base	
106 System Parameters	8 Scan	18 Scan End
107 Open/Close	10 Base	
108 Bind Start	10 Base	109 Bind End
125 RID Pool Processing	8 Scan	18 Scan End
157 DRDS RDS Interface	16 Base	None
159 DRDS Req Site Data	16 Base	None
160 DC Requester	16 Base	None
161 DC Server	16 Base	None
162 DTM Request	16 Base	None
163 DTM Respond	16 Base	None
174 Arch Log CMD Sus Start	3 Base	175 Arch Log CMD Sus End
177 Package Allocation	3 Base	None
183 DRDS RDS/SCC Interface	16 Base	183 DRDS RDS/SCC Interface return
185 READs Data Capture Start		188 READs Data Capture End
213 Drain Lock Wait Start	6 Locking	214 Drain Lock Wait End
215 Claim Count 0 Wait Start	6 Locking	216 Claim Count 0 Wait End
218 Lock Avoidance Summary	6 Locking	None

Table 6. IFCIDs used for SQL Activity reports (continued)

Start IFCID	DB2 trace class & SQL Activity event type	End IFCID
221 Parallel Group Execution	8 Scan	18 Scan End
222 Parallel Group Elapsed Time	8 Scan	18 Scan End
226 Page Latch Contention Start	4 Locking	227 Page Latch Contention End
231 Parallel Group Task Time	8 Scan	18 Scan End
233 Call User Routine	3 Base	233 Call User Routine end
237 Set Current Degree	3 Base	58 End SQL
239 Overflow Package/DBRM	7 Accounting	None
247 SQLDA Data and Input Host Variable Data	5 Host variables data	None
272 Associate Locators	3 Base	58 End SQL
273 Allocate Cursor	3 Base	58 End SQL
305 Table Check Constraint	8 Scan	18 Scan End
324 Function Resolution	3 Base	58 End SQL
325 Trigger Activation	3 Base	58 End SQL

Collecting SQL Activity data with a DB2 trace

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

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

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE (P) CLASS (30) RMID (*) DEST (OPX) PLAN (PMOMDEV) AUTHID (DB2PE)
      IFCID(6,7,8,9,11,12,13,15,16,17,18,22,28,45,53,55,58,59,60,
            61,62,63,64,65,66,68,69,70,71,72,73,74,75,84,85,86,87,
            88,89,92,95,96,97,105,106,107,108,109,125,157,159,160,
            162,163,174,175,177,183,188,214,215,216,221,222,226,227,
            231,233,237,242,272,273,305,324,325) BUFSIZE(512)
```

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

For information regarding DB2 trace data provided by IFCID 350 see “Remark about lengths of SQL statement texts in SQL Activity traces” on page 82.

Creating an SQL Activity trace

Use your ISPF editor to create a JCL job that creates an SQL Activity trace.

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

```

EDIT ---- ANDREW.OMPE.JOBS.SQLTRC----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce an SQL activity trace */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE2,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.SQLT01,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 SQLACTIVITY
0011 TRACE
0012 SUMMARIZEBY (OCCURRENCE)
0013 WORKLOAD (SORT)
0014 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 31. SQL Activity job stream using ISPF/PDF editor

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

SQL Activity trace, summarized by occurrence

Here is an example of an SQL Activity trace, summarized by occurrence.

```

LOCATION: DH1G          OMEGAMON XE for DB2 Performance Expert (V5R3M0)          PAGE: 1-1
GROUP: N/P              SQL ACTIVITY - TRACE                                REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P              ACTUAL FROM: 09/27/13 20:16:13.13
SUBSYSTEM: DH1G
DB2 VERSION: V10

```

SUMMARIZED BY OCCURRENCE, WITH SORT WORKLOAD

```

PRIMAUTH: DB2PE          CONNECT : IMSA          CORRNAME: PYPIB          CONNTYPE: IMS-MPP
ORIGAUTH: LTA0244        PLANNAME: PMOMDEV        CORRNMBR: 0052          THRDTYPE: ALLIED
ENDUSER : 'BLANK'        WSNAME : 'BLANK'          TRANSACT: 'BLANK'

```

TRACE # 1.1 DB2 LUWID: DH1G.DH1G.X'B575F4B5662' ACE ADDRESS: X'1148E8F8'

CICS LUWID: CICS1.CICS1.X'006C264C006C'

```

START TIME: 09/27/13 20:16:13.13 START ELAPSED: 0.000017 START REASON: NEW USER
STOP TIME : 09/27/13 20:16:13.23 STOP ELAPSED : 0.000014 STOP REASON : NEW USER

```

NL	EVENT	TIMESTAMP	ELAP.TIME	TCB	TIME	DETAIL
DBRM						PYPIA
SELECT		20:16:13.13	0.000162			STMT# 408 REOPTIMIZED(NO) KEEP UPDATE LOCKS: N/A ISO(CS) SQLSTATE: 00000 SQLCODE: 0
OPEN		20:16:13.13	0.000016			STMT# 416 CURSOR: CUR_CUSTNAME REOPTIMIZED(NO) KEEP UPDATE LOCKS: NO ISO(CS) SQLSTATE: 00000 SQLCODE: 0
--- WORKLOAD HILITE ---						
SCANS :	2	RECS/SORT:	1.00	I/O REQS:	1	SUSPENDS : N/P EXITS : N/P AMS : N/P
ROWSPROC:	2	WORK/SORT:	1.00	AET/I/O :	0.000005	AET/SUSP : N/P AET/EXIT : N/P AET/AMS : N/P
PAGESCAN:	30	PASS/SORT:	0.00	DATA CAPT:	N/P	RIDS UNUSED: N/P CHECKCON : N/P DEGREE REDUCTION : N/P
LOB_PAGSCAN:	0	LOB_UPD_PAGE :	0			
--- SORT ACTIVITY ---						
MEMBER :	N/P	WORKFILES :	1.00	RECORDS :	1.00	MAX REQUESTED : N/P
TOTAL SORTS :	1	INITIAL WORKFILES :	1.00	RECORD SIZE :	29.00	AVG REQUESTED : N/P
SORT KEYS :	1.00	WORKFILES PARTITIONED :	0.00	KEY SIZE :	8.00	MAX NOT ACQUIRED: N/P
SORT COLUMNS:	3.00	PARTITIONING :	NO	DATA SIZE :	21.00	AVG NOT ACQUIRED: N/P
AET/SORT :	0.000014	PARTITIONING & SORTING:	NO	ROWS DELETED:	0.00	MAX RETURN CODE : 0
SORT TYPE :	ESA	PARTITION TYPE :	NONE	MERGE PASSES:	0.00	
FETCH		20:16:13.13	0.000065			STMT# 425 CURSOR: CUR_CUSTNAME ISO(CS) SQLSTATE: 00000 SQLCODE: 0
UPDATE		20:16:13.15	0.000130			STMT# 483 CURSOR: CUR_CUSTOMER REOPTIMIZED(NO) KEEP UPDATE LOCKS: N/A ISO(CS) SQLSTATE: 00000 SQLCODE: 0
OPEN		20:16:13.15	0.000015			STMT# 493 CURSOR: CUR_WAREHOUSE REOPTIMIZED(NO) KEEP UPDATE LOCKS: NO ISO(CS) SQLSTATE: 00000 SQLCODE: 0

```

--- WORKLOAD HILITE ---
SCANS      : 2 RECS/SORT: 1.00 I/O REQS: 1 SUSPENDS : N/P EXITS : N/P AMS : N/P
ROWSPROC   : 2 WORK/SORT: 1.00 AET/I/O : 0.000006 AET/SUSP : N/P AET/EXIT : N/P AET/AMS : N/P
PAGESCAN   : 30 PASS/SORT: 0.00 DATACAPT: N/P RIDS UNUSED: N/P CHECKCON : N/P DEGREE REDUCTION : N/P
LOB_PAGSCAN: 0 LOB_UPD_PAGE : 0

--- SORT ACTIVITY ---
MEMBER      : N/P WORKFILES : 1.00 RECORDS : 1.00 MAX REQUESTED : N/P
TOTAL SORTS : 1 INITIAL WORKFILES : 1.00 RECORD SIZE : 15.00 AVG REQUESTED : N/P
SORT KEYS   : 1.00 WORKFILES PARTITIONED : 0.00 KEY SIZE : 10.00 MAX NOT ACQUIRED: N/P
SORT COLUMNS: 2.00 PARTITIONING : NO DATA SIZE : 12.00 AVG NOT ACQUIRED: N/P
AET/SORT    : 0.000012 PARTITIONING & SORTING: NO ROWS DELETED: 0.00 MAX RETURN CODE : 0
SORT TYPE   : ESA PARTITION TYPE : NONE MERGE PASSES: 0.00

FETCH      20:16:13.15 0.000094 STMT# 497 CURSOR: CUR_WAREHOUSE SQLSTATE: 00000 SQLCODE: 0
UPDATE     20:16:13.15 0.000071 STMT# 507 CURSOR: CUR_WAREHOUSE ISO(CS) SQLSTATE: 00000 SQLCODE: 0
          REOPTIMIZED(NO) KEEP UPDATE LOCKS: N/A
INSERT     20:16:13.15 0.000096 STMT# 544 ISO(CS) SQLSTATE: 00000 SQLCODE: 0
          REOPTIMIZED(NO) KEEP UPDATE LOCKS: N/A
COMMIT PHASE 1 20:16:13.16 0.004708
COMMIT PHASE 2 20:16:13.16 0.006613

LOCATION: DH1G OMEGAMON XE for DB2 Performance Expert (V5R3M0) PAGE: 1-3
GROUP: N/P SQL ACTIVITY - TRACE REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P TO: NOT SPECIFIED
SUBSYSTEM: DH1G ACTUAL FROM: 09/27/13 20:16:13.13
DB2 VERSION: V10

SUMMARIZED BY THREAD

PRIMAUTH: DB2PE CONNECT : IMSA CORRNAME: PYPIB CONNTYPE: IMS-MPP
ORIGAUTH: LTA0244 PLANNAME: PMOMDEV CORRNMBR: 0052 THRDTYPE: ALLIED
ENDUSER : 'BLANK' WSNM : 'BLANK' TRANSACT: 'BLANK'

TRACE # 1.1 DB2 LUWID: DH1G.DH1G.X'B575F4B56662' ACE ADDRESS: X'1148E8F8'
CICS LUWID: CICS1.CICS1.X'006C264C006C'

START TIME: 09/27/13 20:16:13.13 START ELAPSED: 0.000017 START REASON: NEW USER
STOP TIME : 09/27/13 20:16:13.23 STOP ELAPSED : 0.000014 STOP REASON : NEW USER

EVENT COUNT TOT.ELAPS TOTAL TCB DETAIL
AET/EVENT TCB/EVENT
-----
X'B575F4B56662' 1 0.011792 STMTTYPE COUNT AET/OCCUR TCB/OCCUR COMMITS: 1
0.011792 FETCH 2 0.000080
INSERT 1 0.000096
OPEN 2 0.000015
SELECT 1 0.000162
UPDATE 2 0.000100

LOCATION: DH1G OMEGAMON XE for DB2 Performance Expert (V5R3M0) PAGE: 0-1
GROUP: N/P SQL ACTIVITY - TRACE REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P TO: NOT SPECIFIED
SUBSYSTEM: DH1G ACTUAL FROM: 09/27/13 20:16:13.13
DB2 VERSION: V10 TO: 09/27/13 20:16:13.40

INDEX

SQL TRACE # 1

TRACE START CONNECT CORRNAME CORRNMBR PRIMAUTH ORIGAUTH PLANNAME REQUESTER
SERVER INSTANCE ACE TRACE STARTS
ADDRESS NO PAGE NO
-----
09/27/13 20:16:13.13 IMSA PYPIB 0052 DB2PE LTA0244 PMOMDEV DH1G B575F4B56662 1148E8F8 1.1 1-1

SQL ACTIVITY TRACE COMPLETE

```

Interpreting the result

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

SQL Activity workload highlight block

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

```
OPEN                20:16:13.15    0.000015    STMT# 493 CURSOR: CUR_WAREHOUSE ISO(CS) SQLSTATE: 00000 SQLCODE: 0
                    REOPTIMIZED(NO) KEEP UPDATE LOCKS: NO

--- WORKLOAD HILITE ---
SCANS   : 2 RECS/SORT: 1.00 I/O REQS: 1 SUSPENDS : N/P EXITS : N/P AMS : N/P
ROWSPROC: 2 WORK/SORT: 1.00 AET/I/O : 0.000006 AET/SUSP : N/P AET/EXIT : N/P AET/AMS : N/P
PAGESCAN: 30 PASS/SORT: 0.00 DATACAPT: N/P RIDS UNUSED: N/P CHECKCON : N/P DEGREE REDUCTION : N/P
LOB_PAGSCAN: 0 LOB_UPD_PAGE : 0

--- SORT ACTIVITY ---
MEMBER   : N/P WORKFILES : 1.00 RECORDS : 1.00 MAX REQUESTED : N/P
TOTAL SORTS : 1 INITIAL WORKFILES : 1.00 RECORD SIZE : 15.00 AVG REQUESTED : N/P
SORT KEYS : 1.00 WORKFILES PARTITIONED : 0.00 KEY SIZE : 10.00 MAX NOT ACQUIRED: N/P
SORT COLUMNS: 2.00 PARTITIONING : NO DATA SIZE : 12.00 AVG NOT ACQUIRED: N/P
AET/SORT : 0.000012 PARTITIONING & SORTING: NO ROWS DELETED: 0.00 MAX RETURN CODE : 0
SORT TYPE : ESA PARTITION TYPE : NONE MERGE PASSES: 0.00
```

Verify the average elapsed times for sorts (AET/SORT field. Approximately 0.00001 seconds (0.01 milliseconds) per row sorted is acceptable. Significantly higher values indicate a sort problem.

Sort problems could be caused by any of the following factors:

- Statements with DISTINCT
- Noncorrelated subqueries
- UNION
- Missing index

If your analysis indicates no significant sort problems, the problem could be a scan problem. Look for SQL statements with:

- Nonindexable predicates (WHERE (COL1*COL2) > (COL3*COL4))
- OR connections
- Noncorrelated subqueries
- Use of static SQL rather than dynamic SQL
- Missing index

Remark about lengths of SQL statement texts in SQL Activity traces

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

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

IFCID 350 records SQL statement texts up to a length of 2000000 characters each. However, IFCID 350 data is not shown in SQL Activity reports and traces; this is independent of whether IFCID 350 data is provided by a DB2 trace.

To obtain extended SQL statement texts, perform the following steps:

1. Start a DB2 trace to collect data, using trace class 30, 31, or 32, and specifying IFCID 350, for example:

```
-START TRACE CLASS(30) IFCID(350) DEST(SMF) TDATA(CPU,COR,DIST)
```

2. Create a Record Trace report (refer to Chapter 14, “Record Trace reports,” on page 113 for details).

Use the GLOBAL command and its options to minimize the amount of data that is produced by the Record trace and to filter the required data, similar to the following example:

```
GLOBAL  
  FROM(,12:19:59.99)  
  TO(,12:20:00.93)  
  INCLUDE(IFCID(350))
```

Chapter 10. Using Explain reports to monitor access paths

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

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

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

“Explain reports”

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

“Levels of detail” on page 86

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

“Required authorization” on page 87

Explain reports require SELECT authorization to access certain tables.

“When to use Explain reports” on page 87

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

“Creating an Explain report” on page 88

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

“Explain report example” on page 88

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

“Interpreting the Explain report” on page 93

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

Explain reports

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

Query number

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

- A dynamic SQL EXPLAIN statement was executed with a given query number. The statement can be from DB2 or QMF. If the query number is not specified, DB2 assigns a number. You can get the query number from the PLAN_TABLE.

- The application was bound (or rebound) with EXPLAIN(YES). In this instance, the statement query number is assigned by the precompiler.

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

Statement text

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

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

Plan

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

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

Package

This reports on SQL statements in a package.

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

QMF query

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

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

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

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

Levels of detail

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

Summary

A summary report shows one line for each SQL statement.

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

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

- "Raw" SQL EXPLAIN data as found in the PLAN_TABLE
- Access path data
- Table and table space data

- Index data, this shows:
 - All available indexes for a given table
 - Indexes selected for the access only
- Key data
- Plan and package data
- Host variables data
- Summary page

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

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

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

Index Data

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

No Raw PLAN_TABLE Data

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

Key Distribution

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

Required authorization

Explain reports require SELECT authorization to access certain tables.

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

To explain saved QMF queries, you need SELECT authorization for the following QMF tables:

- Q.OBJECT_DIRECTORY
- Q.OBJECT_DATA

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

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

When to use Explain reports

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

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

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

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

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

Creating an Explain report

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

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

```

EDIT ---- ANDREW.OMPE.JOBS.EXPL1 ----- Columns 001 072
Command ==> Scroll ==> CSR_
***** TOP OF DATA *****
0001 /* JCL to produce an explain report */
0002 // AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAN
0005 // STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 // SYSOUT DD SYSOUT=*
0007 // EXPLAIN DD SYSOUT=*
0008 // DPMOUTDD DD DSN=ANDREW.OMPE.OUT.EXPL1,DISP=OVR
0009 // JOBSUMDD DD SYSOUT=A
0010 //SYSIN DD *
0011 EXPLAIN
0012 PACKAGE (K02EX520.DG0@TPG3.(-2))
0013 SSID(SDA2)
0014 LEVEL(DETAIL)
0015 HOSTVAR(YES)
0016 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 32. Explain JCL

This example produces an Explain detail report for the second generation of a package (DG0@TPG3) on DB2 subsystem SDA2.

Explain report example

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

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

Explain report – page header

```
ACTUAL AT: 02/16/13 15:26:53 OMEGAMON XE DB2 PE (V5.3) PAGE      : 1-1
                                EXPLAIN PACKAGE          DB2 VERSION: V10
LOCATION  : PMODB2A                                USER AUTHID: XRK
SUBSYSTEM: SDA2                                  DETAIL      CURR.SQLID : XRK
```

Explain report – package version details

FPEY0166I PACKAGE DGO@TPG3 IN COLLECTION K02EX520 HAS THE FOLLOWING VERSIONS

PRE-COMP'D EXP GEN VERSION IDENTIFICATION

```
-----
2013-02-16 YES    0 0510_PM81053C
2011-08-16 YES   -1 0510_PM24082C
2011-07-15 YES   -2*OMPE_FINAL
2011-06-14 YES   -3 0510_TEST_3
2011-05-13 YES   -4 0510_TEST_2
2011-04-12 YES   -5 0510_TEST_1
2011-03-11 YES   -6 < VERSION IDENTIFICATION NOT SPECIFIED >
```

```
START VERSION GENERATION NUMBER SPECIFIED:  -2
NUMBER OF VERSION GENERATIONS REQUESTED:     1
```

Explain report – statement details

```
PACKAGE LOCATION  : PMODB2A
PACKAGE COLLECTION: K02EX520
PACKAGE ID        : DGO@TPG3
PACKAGE VERSION ID: OMPE_FINAL
STATEMENT NUMBER  : 1011071
SQL STATEMENT READ FROM SYSIBM.SYSPACKSTMT:
```

```
DECLARE C_DGOYTPG 71 CURSOR WITH HOLD FOR
SELECT LOCATION, COLLID, NAME, CONTOKEN, OWNER, CREATOR, TIMESTAMP, BINDTIME,
QUALIFIER, PKSIZE, AVGSIZE, SYSENTRIES, VALID, OPERATIVE, VALIDATE,
ISOLATION, RELEASE, EXPLAIN, QUOTE, COMMA, HOSTLANG, CHARSET, MIXED, DEC31,
DEFERPREP, SQLERROR, REMOTE, PCTIMESTAMP, IBMREQD, VERSION, PDSNAME, DEGREE,
GROUP_MEMBER, DYNAMICRULES, REOPTVAR, DEFERPREPARE, KEEP_DYNAMIC, PATHSCHEMAS,
TYPE, DBPROTOCOL, FUNCTIONTS, OPTHINT, ENCODING_CCSID, IMMEDIATEWRITE, RELBOUND,
CATENCODE, REMARKS
FROM DGO_SYSPACKAGE
WHERE LOCATION LIKE :HV_LOC71_LOCATION AND COLLID LIKE :HV_LOC71_COLLID AND
NAME LIKE :HV_LOC71_NAME AND VERSION LIKE :HV_LOC71_VERSION
ORDER BY LOCATION, COLLID, NAME, PCTIMESTAMP DESC QUERYNO 001011071
```

```
STATUS      : COMPILED-BOUND USING DEFAULTS FOR INPUT VARIABLES
ISOLATION: UNCOMMITTED READ / FROM SYSPACKAGE
```

```
-----
EXPLAIN TABLE: PMDEV52.DSN_STATEMNT_TABLE -----
EXPLAIN_TIME    : 2013-02-16-15.24.35.310181
```

```
PROGNAME (Package): DGO@TPG3          , COLLID      : K02EX520
VERSION          : N/A                , APPLNAME (Plan) : N/P
QUERYNO         : 1011071            , SECTNOI       : N/A
STMT_ENCODE     : U - Unicode         , STMT_TYPE     : SELECT

PROCMS (Cost MS) : 1                  , COST_CATEGORY  : B - Default *
PROCSU (Cost SU) : 18                 , REASON (Category) : HOST VARIABLES
TOTAL_COST      : N/A                 , GROUP_MEMBER   : SDA2

COST_CATEGORY    : B - Cost estimate using default values (Details in REASON)
```

Explain report – PLAN_TABLE details

```
EXPLAIN TABLE: PMDEV52.PLAN_TABLE -----
BIND_TIME      : 2013-02-16-15.24.35.310181
TIMESTAMP      : 2013-02-16-15:24:35.34

PROGNAME (Package): DG00TPG3      , COLLID      : K02EX520
VERSION        : OMPE_FINAL      , APPLNAME (Plan) : N/P
QUERYNO        : 1011071        , SECTNOI       : N/A
QBLOCKNO       : 1              , PARENT_QBLOCKNO : 0
PLANNO         : 1              , PARENT_PLANNO   : 0
MIXOPSEQ       : 0              , QBLOCK_TYPE    : SELECT

TNAME (Table)   : SYSPACKAGE      , CREATOR (Table) : SYSIBM
TABNO (Table)   : 1              , CORRELATION_NAME : N/P
TABLE_TYPE      : T - Table       , CTEREF         : 0
TABLE_ENCODE     : U - Unicode    , TABLE_MCCSID   : 1208
TABLE_SCCSID     : 367           , TABLE_DCCSID   : 1200
TSLOCKMODE      : N - No lock *   , GROUP_MEMBER    : SDA2

ACCESSTYPE      : I - Index scan   , PRIMARY_ACCESTYPE: BLANK
ACCESSNAME (Index): DSNKKX01      , ACCESSCREATOR   : SYSIBM
MATCHCOLS       : 1              , INDEXONLY       : NO
METHOD (Join)    : 0 - First table , JOIN_DEGREE     : 0
JOIN_TYPE        : b - INNER or NO , MERGN           : N/A
MERGE_JOIN_COLS  : 0              , MERGC           : N/A
PREFETCH        : S - Sequential  , PAGE_RANGE      : NO
WHEN_OPTIMIZE    : b - At bind time, ACCESS_DEGREE   : 0
COLUMN_FN_EVAL   : BLANK          , ROUTINE_ID      : 0
HINT_USED        : N/P            , OPTHINT         : N/P
SCAN_DIRECTION   : N/A

SORTN_PGROUP_ID : 0              , SORTN_UNIQ      : NO , SORTC_UNIQ : NO
SORTC_PGROUP_ID : 0              , SORTN_JOIN      : NO , SORTC_JOIN  : NO
ACCESS_PGROUP_ID : 0              , SORTN_ORDERBY   : NO , SORTC_ORDERBY: NO
JOIN_PGROUP_ID   : 0              , SORTN_GROUPBY   : NO , SORTC_GROUPBY: NO

REMARKS          : N/P            , STMTTOKEN       : N/P
PARALLELISM_MODE : BLANK          , BIND_EXPLAIN_ONLY : N/A
EXPANSION_REASON  : N/A

TSLOCKMODE       : N - No lock (UR isolation)
```

Explain report – access path

THE ACCESS PATH CHOSEN BY DB2 AT 15:24:35.3 ON 2013-02-16

```
+-----+
| MATCHING INDEX SCAN WITH SCAN OF REFERENCED DATA PAGES
| NUMBER OF MATCHING COLUMNS: 1 - THE INDEX HAS 4 COLUMNS
| NON-CLUSTERED INDEX SCAN WILL BE USED
| PURE SEQUENTIAL PREFETCH WILL BE PERFORMED
| PAGE RANGE SCAN WILL NOT BE USED
+-----+
```

Explain report – index details

```
INDEX: SYSIBM.DSNKKX01 -----
STATTIME: 2013-02-16-15.02.47.358366
CREATED  : 0001-01-01-00.00.00.000000    ALTERED: 2003-09-21-23.30.17.962937
FULL KEY CARD: 1093,PAGES : 20,LEVELS: 2,CLUSTERING: Y
1"ST KEY CARD: 1,SPACE : 196.608K,UNIQUE: YES,CLUSTERED: N
INDEX TYPE   : 2,PGSIZE: 4096,BFPOOL: BPO,DB.NAME : DSNDB06
CLUSTERRATIO : 83.5316%,ERRULE: NO,CLRULE: NO,IXSPACE : DSNKKX01
MAX.PIECESIZE: 0,COPY : NO,COPYLRN: X'000000000000'
```

Explain report – key column details

KEY NO. COLUMN NAME	COL.TYPE COL.STAT	LNG LENGTH2	NULL	KEY CARD. ORDER TYPESCHEMA	LOW2KEY TYPENAME CREATED	HIGH2KEY	KEY USED
1 LOCATION	VARCHAR	128	NO	1 ASC. 0 SYSIBM	X'40404040 VARCHAR 0001-01-01-00.00.00.000000	X'40404040	<===
2 COLLID	VARCHAR	128	NO	60 ASC. 0 SYSIBM	X'4144424C VARCHAR 0001-01-01-00.00.00.000000	X'55545255	
3 NAME	VARCHAR	128	NO	504 ASC. 0 SYSIBM	X'41444232 VARCHAR 0001-01-01-00.00.00.000000	X'535F5550	
4 VERSION	VARCHAR	122	NO	87 ASC. 0 SYSIBM	X'31404040 VARCHAR 0001-01-01-00.00.00.000000	X'56385231	

Explain report – table details

TABLE: SYSIBM.SYSPACKAGE -----
 STATTIME: 2013-02-16-15.02.47.358366, TB TYPE : TABLE
 CREATED : 1985-04-01-00.00.00.000000, ALTERED : 2003-09-21-23.30.17.962937
 ROWS : 1093, COLUMNS : 47, ROWLENGTH: 3894, EDIT PROC.:
 % PAGES : 90, DBASE ID: 6, AUDITING : NONE, VALIDPROC.:
 ACT.PAGES: 65, TABLE ID: 128, STATUS : COMPX, TABCREATOR: SYSIBM
 TAB.STAT.: , ENC.SCHEME: UNICODE

Explain report – table space details

TABLESPACE: DSNDB06.SYSPKAGE -----
 NAME : SYSPKAGE , DATABASE : DSNDB06
 CREATOR : SYSIBM , CREATED BY: SYSIBM
 CREATED : 0001-01-01-00.00.00.000000, ALTERED : 2003-09-21-23.30.17.962937
 STATTIME : 2013-02-16-15.02.47.358366,
 ACTIVE PGS: 8640, DBASE ID : 6, TS STATUS : A, TS TYPE : NORMAL
 PAGE SIZE : 4KB, OBJ ID : 120, ERASERULE : NO, STORGROUP :
 SPACE : 22.118M, PAGESET ID: 121, CLOSERULE : NO, BUF.POOL : BP0
 DS SIZE : 0, OLD VERS : 2, LOCKPART : N/A, ENC.SCHEME: UNICODE
 LOCKMAX : SYSTEM, CUR VERS : 2, LOCKRULE : ANY, SBCS CCSID: 367
 TABLES/TS : 7, PARTITIONS: UNPAR, LOG : YES, DBCS CCSID: 1200
 MAXROWS : 255, SEG SIZE : 4, IMPLICIT : NO, MIX. CCSID: 1208
 AVG ROWLEN: 235,

Explain report – host variables details

HOST VAR. TYPE	LENGTH	IND.	HOST VARIABLE NAME
VAR. CHARACTER	16	NO	HV_LOC71_LOCATION
VAR. CHARACTER	18	NO	HV_LOC71_COLLID
VAR. CHARACTER	8	NO	HV_LOC71_NAME
VAR. CHARACTER	64	NO	HV_LOC71_VERSION

Explain report – language and compile details

 LOCATION : PMODB2A
 COLLECTION ID: K02EX520
 PACKAGE ID : DGO@TPG3
 VERSION ID : OMPE_FINAL
 CONSIST.TOKEN: X'18C8F8D602A50FF6'
 PDSNAME : D010MPE.BASE51.TK02DBRM

OWNER : PMDEV52	QUOTE : APOSTROPHE
CREATOR : XRK	COMMA : PERIOD
BIND DATE : 2013-02-16	HOSTLANG : ASSEMBLER
BIND TIME : 15.24.35.310181	CHARSET : ALPHANUMERIC

```

CREATE DATE   : 2013-02-16          MIXED           : NO
CREATE TIME   : 15.24.23.620690     DEC31          : NO
QUALIFIER     : PMDEV52             DATA CURRENCY: INHIBIT BLOCKING
BASE SIZE     : 5048                SQLERROR       : NOPACKAGE
AVERAGE SIZE  : 49764              SOURCE         : DBRM
SYSENTRIES    : 0                  PRECOMP. DATE: 2011-07-15
SQL STATEMENT : 17                 PRECOMP. TIME: 13.20.55.105599
VALIDATE      : BIND               VALID           : YES
ISOLATION     : UNCOMMITTED READ    OPERATIVE      : YES
RELEASE       : CHECK PLAN          REOPTIMIZAT.   : NO
DEGREE        : 1                  DEFERPREPARE   : INHERITED FROM PLAN
KEEP DYNAMIC  : DRDA               DDF PROTOCOL   : INHERITED FROM PLAN
TYPE OF PACK.: BIND PACKAGE        OPT_HINT_ID    :
FNCT.RESOLVED: 2013-02-16-15.24.35.298423
      ....5....0....5....0....5....0....5....0....5....0....5....0....5...
PATH:
=====

```

Explain report – statement details (repeated for second statement step)

```

PACKAGE LOCATION : PMODB2A
PACKAGE COLLECTION: K02EX520
PACKAGE ID       : DGO@TPG3
PACKAGE VERSION ID: OMPE_FINAL
STATEMENT NUMBER : 1011071
SQL STATEMENT READ FROM SYSIBM.SYSPACKSTMT:

```

```

DECLARE C_DGOYTPG_71 CURSOR WITH HOLD FOR
SELECT LOCATION, COLLID, NAME, CONTOKEN, OWNER, CREATOR, TIMESTAMP, BINDTIME,
QUALIFIER, PKSIZE, AVGSIZE, SYSENTRIES, VALID, OPERATIVE, VALIDATE,
ISOLATION, RELEASE, EXPLAIN, QUOTE, COMMA, HOSTLANG, CHARSET, MIXED, DEC31,
DEFERPREP, SQLERROR, REMOTE, PCTIMESTAMP, IBMREQD, VERSION, PDSNAME, DEGREE,
GROUP_MEMBER, DYNAMICRULES, REOPTVAR, DEFERPREPARE, KEEP DYNAMIC, PATHSCHEMAS,
TYPE, DBPROTOCOL, FUNCTIONTS, OPTHINT, ENCODING_CC SID, IMMEDIATEWRITE, RELBOUND,
CATENCODE, REMARKS
FROM DGO_SYSPACKAGE
WHERE LOCATION LIKE :HV_LOC71_LOCATION AND COLLID LIKE :HV_LOC71_COLLID AND
NAME LIKE :HV_LOC71_NAME AND VERSION LIKE :HV_LOC71_VERSION
ORDER BY LOCATION, COLLID, NAME, PCTIMESTAMP DESC QUERYNO 001011071

```

```

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

```

Explain report – PLAN_TABLE (for second statement step)

```

EXPLAIN TABLE: PMDEV52.PLAN_TABLE -----
BIND TIME      : 2013-02-16-15.24.35.310181
TIMESTAMP      : 2013-02-16-15:24:35.34

PROGNAME (Package): DGO@TPG3          , COLLID           : K02EX520
VERSION          : OMPE_FINAL          , APPLNAME (Plan)  : N/P
QUERYNO         : 1011071             , SECTNOI          : N/A
QBLOCKNO        : 1                   , PARENT_QBLOCKNO  : 0
PLANNO          : 2                   , PARENT_PLANNO    : 0
MIXOPSEQ        : 0                   , QBLOCK_TYPE      : SELECT

TNAME (Table)   : N/P                 , CREATOR (Table)  : N/P
TABNO (Table)   : 0                   , CORRELATION_NAME : N/P
TABLE_TYPE      : N/P                 , CTEREF           : 0
TABLE_ENCODE    : BLANK               , TABLE_MCCSID    : 0
TABLE_SCESID    : 0                   , TABLE_DCCSID    : 0
TSLOCKMODE      : BLANK               , GROUP_MEMBER     : SDA2

ACCESSTYPE      : BLANK               , PRIMARY_ACCESSTYPE: BLANK
ACCESSNAME (Index): N/P              , ACCESSCREATOR    : N/P
MATCHCOLS      : 0                   , INDEXONLY        : NO

```



```

METHOD (Join)      : 3 - Sort needed      , JOIN_DEGREE      : 0
JOIN_TYPE          : b - INNER or NO      , MERGN            : N/A
MERGE_JOIN_COLS    : 0                    , MERGC            : N/A
PREFETCH           : BLANK                 , PAGE_RANGE       : NO
WHEN_OPTIMIZE      : b - At bind time     , ACCESS_DEGREE    : 0
COLUMN_FN_EVAL     : BLANK                 , ROUTINE_ID       : 0
HINT_USED          : N/P                   , OPTHINT          : N/P
SCAN_DIRECTION     : N/A

SORTN_PGROUP_ID    : 0                    , SORTN_UNIQ       : NO , SORTC_UNIQ       : NO
SORTC_PGROUP_ID    : 0                    , SORTN_JOIN       : NO , SORTC_JOIN       : NO
ACCESS_PGROUP_ID   : 0                    , SORTN_ORDERBY    : NO , SORTC_ORDERBY    : YES
JOIN_PGROUP_ID     : 0                    , SORTN_GROUPBY    : NO , SORTC_GROUPBY    : NO

REMARKS            : N/P                   , STMTOKEN         : N/P
PARALLELISM_MODE   : BLANK                 , BIND_EXPLAIN_ONLY : N/A
EXPANSION_REASON    : N/A

```

Explain report – access path (for second statement step)

THE ACCESS PATH CHOSEN BY DB2 AT 15:24:35.3 ON 2013-02-16

```

+-----+
| ADDITIONAL SORT FOR ORDER BY |
| PAGE RANGE SCAN WILL NOT BE USED |
+-----+

```

Explain summary report

REPORT ON: 02/16/13 15:26:53 OMEGAMON XE DB2 PE (V5.3) PAGE : SUMMARY
EXPLAIN SUMMARY REPORT USER AUTHID: XRK

THE FOLLOWING 1 EXPLAIN REQUESTS WERE PROCESSED: PAGE NO

```

1:   SDA2 PACKAGE : K02EX520 .DGO@TPG3
      DETAIL REPORT REQUESTED
      DBRM/PACK STMT TYP
      FOR A LIST OF VERSIONS, PLEASE REFER TO PAGE 1-1
      DGO@TPG3 1011020 P MATCHING INDEX SCAN(2/4)-DATA PAGES 1-3
      DGO@TPG3 1011021 P S TABLE SPACE SCAN-NO INDEX WILL BE USED 1-7
      DGO@TPG3 1011021 P ADDITIONAL SORT FOR ORDER BY 1-10
      DGO@TPG3 1011070 P MATCHING INDEX SCAN(3/4)-DATA PAGES 1-12
      DGO@TPG3 1011071 P S MATCHING INDEX SCAN(1/4)-DATA PAGES 1-16
      DGO@TPG3 1011071 P ADDITIONAL SORT FOR ORDER BY 1-20

```

Interpreting the Explain report

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

For a full discussion of the DB2 catalog information shown by the Explain report, see the *IBM Knowledge Center*.

The information reported varies depending on the type of report, the level of detail requested, and whether data is available to report. The example shown in the previous section shows an Explain at package level. The package explained contains SQL statements. The statement QUERYNO=1011071 has two statement steps (Select with index for PLANNO=1 and sort for ORDER BY for PLANNO=2).

Page header

The page header (“Explain report – page header” on page 89) is printed at the top of each report page and shows general information about the report, including the type of report, subsystem identifier, DB2 version, and the objects reported.

Package version

The Package Version block of the report (“Explain report – package version details” on page 89) shows the package versions available and the versions and generations reported.

Statement details

The Statement Details block of the report (“Explain report – statement details” on page 89) shows the complete plan or package identification details, including:

- Location
- Name
- Statement number and text
- Other statement-related information

It also shows the status and isolation level, from SYSIBM.SYSPLAN or SYSIBM.SYSPACKAGE or SYSIBM.SYSPACKSTMT, and cost information.

PLAN_TABLE information

This shows the contents of the PLAN_TABLE (“Explain report – PLAN_TABLE details” on page 90). The report block starts with EXPLAIN TABLE and has two block columns. The report field labels are the full PLAN_TABLE column names. If the report field value is a long name or a long value, it is marked with an asterisk (*) at the end of the report field value, and reported at the end of the block in its full length.

The plan table is searched by using the bind time of the package. There might be multiple occurrences of the package with different bind times. For example, the DB2 system catalog might contain the latest package that is created by the DB2 command BIND or REBIND. It might also contain a former package version that is activated by the DB2 command REBIND SWITCH.

Access path

The Access Path block of the report (“Explain report – access path” on page 90) shows, for each step in an SQL statement, the access path used by DB2 to satisfy the statement step. This shows:

Use of index

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

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

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

No index

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

Index on COL4

In this instance, the number of Getpage operations depends on the number of rows matching the predicate and their distribution, and the number of index pages. For example, if rows matching the predicate are spread over

six data pages and pointers are spread over two index pages, the number of Getpage operations for the SELECT is eight.

Unique index on COL4 and COL5

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

Index only scan

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

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

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

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

You can define multiple indexes on a base table. However, weigh the performance gain with the additional workload required for the database manager to update the indexes when data changes in the table. Generally, multiple indexes are useful for tables that are often queried and seldom updated.

You can use the Accounting reports to derive the Getpage/SQL ratios for plans and packages. This is the total number of Getpage operations divided by the total number of SQL SELECT, INSERT, UPDATE, and FETCH statements. The value of the ratio depends on the type of application and the DB2 environment, but you might look for applications with ratios greater than five.

Prefetch

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

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

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

Direct row access

Direct row access allows DB2 to obtain access to a row directly through the ROWID column. If an application selects a row from a table that contains a ROWID column, the row ID value implicitly

contains the location of the row. If you use that row ID value in the search condition of subsequent SELECTs, DB2 might be able to navigate directly to the row.

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

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

Index data

The Index Data block of the report ("Explain report – index details" on page 90) shows data derived from the SYSIBM.SYSINDEXES table. This is only shown if an index is used in the access plan. You can use the Information in SYSINDEXES to compare the available indexes on a table to determine which one is the most efficient for a query.

Key column data

The Key Column Data block of the report ("Explain report – key column details" on page 91) shows index key information that is derived from the SYSIBM.SYSKEYS and SYSIBM.SYSCOLUMNS tables. This is only shown when an index is used in the access plan.

Table The Table Data block of the report ("Explain report – table details" on page 91) shows information that is derived from SYSIBM.SYSTABLES.

Table space

The Table Space Data block of the report ("Explain report – table space details" on page 91) shows information that is derived from SYSIBM.SYSTABLESPACE.

Host variables data

When host variables are used in a statement, OMEGAMON XE for DB2 PE can show details about each variable ("Explain report – host variables details" on page 91). A host variable can be either a variable in a host language (such as a PL/I variable, C variable, Fortran variable, a COBOL data item, or Assembler language storage area) or a host language construct that was generated by an SQL precompiler from a variable declared using SQL extensions. A host variable can be an output value that is returned to the application by DB2 or an input to DB2.

Summary

The summary report ("Explain summary report" on page 93) shows the package name, SQL statement number, type, and access path for each statement reported. When the summary is appended to a full report, the statement numbers are also cross-referenced to the report page where the full entry can be found.

Chapter 11. Locking Activity reports and traces

The locking activity provides various levels of detail about concurrency control within DB2 in the form of reports, traces, and a file data set.

It shows information about:

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

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

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

Locking activity traces (also referred to as locking traces) provide information about lock events as they occur. Locking traces differ in the level of detail as follows:

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

“When to use Locking reports” on page 98

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

“Input for Locking reports” on page 98

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

“Collecting Locking Activity data with a DB2 trace” on page 99

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

“Creating a Detail Locking trace” on page 99
Use your ISPF editor to create a JCL job that creates a Detail Locking trace.

When to use Locking reports

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

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

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

Input for Locking reports

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

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

Table 7. IFCIDs used for Locking reports

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

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

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

To maintain consistency in a data sharing environment, DB2 IFC generates an IFCID 251, 257, or 259, which records, or notifies on, a physical lock (P-lock) on a page set, partition, or page.

Whenever a suspension occurs, DB2 IFC generates an IFCID 44, 213, 215, or 226 record along with a matching IFCID record that specifies the reason for resume,

regardless of how the suspension was generated. The matching IFCID record is an IFCID 45 for IFCID 44, IFCID 214 for IFCID 213, IFCID 216 for IFCID 215, and IFCID 227 for IFCID 226. Also, the DB2 trace produces IFCID 211 records for every claim request and IFCID 212 records for every drain request.

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

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

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

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

IFCID 21 is the base for recording the detail lock requests.

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

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

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

Collecting Locking Activity data with a DB2 trace

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

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

Alternatively, you can start a DB2 trace as follows:

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

Creating a Detail Locking trace

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

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


```

EDIT ---- ANDREW.OMPE.JOBS.LOKTRC----- Columns 001 072
Command ==> Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce a Locking detail trace */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.LOCKDT,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.LOCK1,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 LOCKING
0011 TRACE
0012 LEVEL (DETAIL)
0013 SCOPE (MEMBER)
0014 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 33. Locking activity job stream using ISPF/PDF editor

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

Example of a detail Locking trace

The following example shows a detail Locking trace.

LOCATION: USIBMSYSTD2	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 1-4
GROUP: USIBMSYSTD2	LOCKING TRACE - DETAIL	REQUESTED FROM: NOT SPECIFIED
MEMBER: DB2U		TO: NOT SPECIFIED
SUBSYSTEM: DB2U		ACTUAL FROM: 08/14/13 19:32:57.03
DB2 VERSION: V10	SCOPE: MEMBER	PAGE DATE: 08/14/13

PRIMAUTH	CORRNAME	CONNTYPE	EVENT	TIMESTAMP	--- L O C K R E S O U R C E ---</th <th>EVENT SPECIFIC DATA</th>	EVENT SPECIFIC DATA
ORIGAUTH	CORRNMBR	INSTANCE	RELATED	TIMESTAMP	TYPE NAME	
PLANNAME	CONNECT		EVENT			
USER005	NOPID	IMS-MPP	19:32:57.03444843	LOCK	PAGESET DB =DSNDB01	DURATION=ALLOCATN STATE=IS XES PROP=N
USER005	0002	A9BA4BA9A1A6	REQUEST		OB =SCT02	RSN CODE=X'20' RTNCD= 4 XES FORC=N
NOPID	IMSA					NMODIFY GLOBAL L-LOCK HASH =X'00000BC0'
			19:32:57.05134612	LOCK	DATAPAGE DB =DSNDB01	DURATION=MANUAL STATE=S XES PROP=N
			REQUEST		OB =SCT02 PAGE=X'0000009A'	RSN CODE= 0 RTNCD= 0 XES FORC=N
						NMODIFY GLOBAL L-LOCK PARENT =X'03E490CB' HASH =X'00019A5F'
			19:32:57.07211528	LOCK	SKCT PLAN=NOPID	DURATION=INTEREST STATE=IS XES PROP=Y
			REQUEST			RSN CODE= 0 RTNCD= 0 XES FORC=N
						NMODIFY GLOBAL P-LOCK HASH =X'11969789'
			19:32:57.07269878	UNLOCK	DATAPAGE DB =DSNDB01	DURATION=MANUAL STATE=S XES PROP=N
			19:32:57.05134612	REQUEST	OB =SCT02 PAGE=X'0000009A'	RSN CODE= 0 RTNCD= 0 XES FORC=N
						NMODIFY GLOBAL L-LOCK HASH =X'00019A5F'
			19:32:58.89454559	LOCK	OPENLOCK DB =TPCCE1	DURATION=MANUAL STATE=X XES PROP=Y
			SUSPEND		OB =TCUST000	ORIG.RSN=INTER SYSTEM XES FORC=N
						NMODIFY GLOBAL L-LOCK PARENT =X'01020006' HASH =X'00002080'
			19:32:58.89622303	LOCK	OPENLOCK DB =TPCCE1	SUSP.TIME =0.001677 LOCAL CONTENTION=N
			19:32:58.89454559	RESUME	OB =TCUST000	DURATION =MANUAL LATCH CONTENTION=N
						STATE =X IRLM QUEUED REQ =N
						RESUME RSN=NORMAL GLOBAL CONT. =N
						XES PROP =Y NOTIFY MSG SENT =N
						XES FORC =N
						XES ASYN =Y RETAINED LOCK =Y
						NMODIFY GLOBAL L-LOCK PARENT =X'01020006' HASH =X'00002080'
			19:32:58.89633406	LOCK	OPENLOCK DB =TPCCE1	DURATION=MANUAL STATE=X XES PROP=Y
			REQUEST		OB =TCUST000	RSN CODE= 0 RTNCD= 0 XES FORC=N
						NMODIFY GLOBAL L-LOCK HASH =X'00002080'

19:32:58.89738515	LOCK REQUEST	GBP CONV	DB =TPCCE1 OB =TCUST000	DURATION=MANUAL STATE=S RSN CODE= 0 RTNCD= 0 NMODIFY GLOBAL L-LOCK HASH =X'00000403'	XES PROP=Y XES FORC=N XES ASYN=N
19:32:58.89859378	LOCK REQUEST	P/P PLCK	DB =TPCCE1 OB =TCUST000 BPID=BP0	DURATION=INTEREST STATE=S RSN CODE= 0 RTNCD= 0 NMODIFY GLOBAL P-LOCK CACHE =S HASH =X'00000402'	XES PROP=Y XES FORC=N XES ASYN=N
19:32:58.89876712	P-LOCK REQUEST	P/P PLCK	DB =TPCCE1 OB =TCUST000 BPID=BP0	REQUEST=LOCK OBJECT=TABLESPACE MEMBER =N/A REQUESTED STATE=S OLD STATE=NH OLD CACHED STATE=NH NEW STATE=S NEW CACHED STATE=S CONDITIONAL RESTART NONMODIFY	
19:32:58.89943440	UNLOCK REQUEST	GBP CONV	DB =TPCCE1 OB =TCUST000	DURATION=MANUAL STATE=X'00' RSN CODE= 0 RTNCD= 0 NMODIFY LOCAL L-LOCK HASH =X'00000000'	XES PROP=Y XES FORC=N XES ASYN=N
19:32:58.89633406	UNLOCK REQUEST	OPENLOCK	DB =TPCCE1 OB =TCUST000	DURATION=MANUAL STATE=X RSN CODE= 0 RTNCD= 0 NMODIFY GLOBAL L-LOCK HASH =X'00002080'	XES PROP=Y XES FORC=N XES ASYN=N
19:32:59.93528409	LOCK AVOIDANCE	DATAPAGE	DB =TPCCE1 OB =TCUST000 PAGE=X'00000002'		
19:33:02.09291456	LOCK REQUEST	DATAPAGE	DB =TPCCE1 OB =TSTCK000 PAGE=X'00003A55'	DURATION=MANUAL STATE=U RSN CODE= 0 RTNCD= 0 NMODIFY GLOBAL L-LOCK PARENT =X'03E494C8' HASH =X'000055FA'	XES PROP=Y XES FORC=N XES ASYN=N
19:33:02.09579509	CHANGE REQUEST	DATAPAGE	DB =TPCCE1 OB =TSTCK000 PAGE=X'00003A55'	DURATION=COMMIT STATE=X RSN CODE= 0 RTNCD= 0 MODIFY GLOBAL L-LOCK HASH =X'000055FA'	XES PROP=Y XES FORC=N XES ASYN=N
19:33:02.35630903	CLAIM RELEASE	PAGESET	DB =TPCCE1 OB =TSTCK000	CLASS=WRITE RSN CODE= 0 RTNCD= 0	
19:33:02.35708056	LOCK SUMMARY	ALL PAGESETS		MAX PAGE & ROW LOCKS= 11 SHARED ESCAL= 0 EXCLUS.ESCAL= 0	LOCKAV=YES
		PAGESET	DB =DSNDB01 OB =SCT02	MAX PAGE & ROW LOCKS= 1 TABLESPACE TYPE=UNSEGMENTED MAX STATE=IS NO ESCALATIONS	LOCKAV=NO SIZE=PAGE
		PAGESET	DB =TPCCE1 OB =TCUST000	MAX PAGE & ROW LOCKS= 1 TABLESPACE TYPE=UNSEGMENTED MAX STATE=IS NO ESCALATIONS	LOCKAV=YES SIZE=PAGE
		PAGESET	DB =TPCCE1 OB =TSTCK000	MAX PAGE & ROW LOCKS= 3 TABLESPACE TYPE=UNSEGMENTED MAX STATE=IX NO ESCALATIONS	LOCKAV=NO SIZE=PAGE
19:33:02.36281765	UNLOCK REQUEST	N/P	N/P	DURATION=COMMIT STATE=X'00' RSN CODE=X'02' RTNCD= 4 NMODIFY LOCAL L-LOCK HASH =X'00000000'	XES PROP=Y XES FORC=N XES ASYN=N

Chapter 12. I/O Activity reports

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

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

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

Cross-invalidation is necessary to maintain coherency of data within a DB2 data sharing group. It happens when a group member updates a data page and writes that page to the group buffer pool. All members that have this data page cached in their buffer pools are notified that the page was invalidated.

A high level of cross-invalidation affects performance because it effectively reduces the amount of buffer pool space available to the system. When an invalidated page is required, it must be refreshed in the buffer pool, either by retrieving the page from the group buffer pool or from a hard disk drive.

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

“When to use I/O Activity reports”

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

“Input for I/O Activity reports” on page 104

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

“Collecting I/O Activity data with a DB2 trace” on page 104

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

“Creating an I/O Activity trace” on page 104

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

When to use I/O Activity reports

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

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

Input for I/O Activity reports

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

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

Table 8. IFCIDs used for I/O Activity reports

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

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

Collecting I/O Activity data with a DB2 trace

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

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

Alternatively, you can start a DB2 trace as follows:

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

Creating an I/O Activity trace

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

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

```

EDIT ---- ANDREW.OMPE.JOBS.IOSUMM----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce an I/O Activity Report */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.IOA,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.IOSUMM,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 IOACTIVITY
0011 REPORT
0012 EXEC
0013 ***** ***** 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 34. I/O Activity job stream using ISPF/PDF editor

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

Example of a summary I/O Activity report

The following example shows a summary I/O Activity report.

LOCATION: DSNCAT GROUP: DSNCAT MEMBER: V71B SUBSYSTEM: V71B DB2 VERSION: V10			OMEGAMON XE for DB2 Performance Expert (V5R3M0) I/O ACTIVITY REPORT - SUMMARY			PAGE: 2-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 08/14/13 19:32:54.82 TO: 08/14/13 19:56:28.53		
BUFFER POOL	TOTALS	AET	EDM POOL	CT/PT/DBD REFERENCES	NOT IN EDM POOL	AET	AVG LEN (BYTES)	
TOTAL I/O REQUESTS	51	0.019885	CURSOR TABLE - HEADER	1	2	0.131381	7656.00	
			CURSOR TABLE - DIRECTORY	1	1	0.000568	228.00	
TOTAL READ I/O REQUESTS	51	0.019885	CURSOR TABLE - RDS SECTION	9	9	0.022214	1475.11	
NON-PREFETCH READS	51		-- TOTAL PLANS -----	11	12	0.030170	1923.64	
PREFETCH READS								
WITHOUT I/O	0		PACKAGE TABLE - HEADER	0	0	N/C	0.00	
WITH I/O	0		PACKAGE TABLE - DIRECTORY	0	0	N/C	0.00	
PAGES READ	0		PACKAGE TABLE - RDS SECTION	0	0	N/C	0.00	
PAGES READ / SUCC READ	0.00		-- TOTAL PACKAGES --	0	0	N/C	0.00	
TOTAL WRITE REQUESTS	0	N/C	DATABASE DESCRIPTORS	0	0	N/C	0.00	
SYNCHRONOUS WRITES	0	N/C						
COUPLING FACILITY CASTOUTS	0	N/C						
PAGES WRITTEN PER WRITE	0.00							
ASYNCHRONOUS WRITES	0	N/C						
COUPLING FACILITY CASTOUTS	0	N/C						
PAGES WRITTEN PER WRITE	0.00							
ACTIVE LOG	TOTALS	AET	ARCHIVE LOG	TOTALS	AET	BOOTSTRAP DATASET	TOTALS	AET
TOTAL WAITS	22	0.018384				TOTAL WAITS	25	0.051894
READ REQUESTS	0	N/C	READ REQUESTS	0	N/C	READ REQUESTS	13	0.078868
			DASD READ	0	N/C			
			TAPE READ	0	N/C			
WRITE REQUESTS	22	0.018384	OFFLOAD REQUESTS	0	N/C	WRITE REQUESTS	12	0.022673
CONT. CI / WRITE	1.00		BLOCKS / OFFLOAD	0.00				
OTHER WAITS	0	N/C	OTHER WAITS	0	N/C	CROSS-INVALIDATION ACTIVITY	TOTALS	
ALLOCATE	0	N/C	ALLOCATE	0	N/C			
DEALLOCATE	0	N/C	DEALLOCATE	0	N/C			
OPEN	0	N/C	OPEN	0	N/C	SYNCHRONOUS READS		27
CLOSE	0	N/C	CLOSE	0	N/C	REFRESHED FROM GROUP BPOOL		27
			HSM RECALL	0	N/C	REFRESHED FROM DASD		0
			CATALOG LOCATE	0	N/C			
			MULTI-DATA SET TAPE	0	N/C	SEQUENTIAL PREFETCHES		0
			TAPE VOL POSITIONING	0	N/C	REFRESHED FROM GROUP BPOOL		0
			WTOR ISSUED	0	N/C	REFRESHED FROM DASD		0
			DATA SET UNAVAILABLE	0	N/C			

	PHYSICAL UNIT UNAV.	0	N/C
	RDR SERV.UNAVAILABLE	0	N/C
I/O ACTIVITY REPORT COMPLETE			

Chapter 13. Utility Activity reports

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

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

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

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

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

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

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

- Bind activity shows:
 - BIND
 - BIND PACKAGE
 - REBIND
 - REBIND PACKAGE

- FREE PLAN
- FREE PACKAGE
- FREE REMOTE PACKAGE
- BIND CONNECT
- CONNECT RESET

- Utility activity

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

“When to run Utility Activity reports”

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

“Input for Utility Activity reports”

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

“Collecting Utility Activity data with a DB2 trace” on page 109

To create an Utility Activity report, you must collect data by using a DB2 trace.

“Creating a Utility Activity trace” on page 110

Use your ISPF editor to create a JCL job that creates a Utility Activity trace.

When to run Utility Activity reports

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

Input for Utility Activity reports

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

The following table shows the utility events that can be collected, together with the trace classes and IFCIDs used.

Table 9. IFCIDs used for Utility Activity reports

DB2 trace type	DB2 trace class	DB2 IFCID	Record type
Accounting	1	3	Accounting data
Bind Events			
Performance	3	22	Minibinds generated by DB2 at bind prepare time
		63	SQL statement to be parsed
		177	Package allocation
Performance	10	108	Begin bind or rebind plan/package
		109	End bind or rebind plan/package
		110	Begin free plan/package
		111	End free plan/package
Performance	16	183	Requesting agent data
Utilities			

Table 9. IFCIDs used for Utility Activity reports (continued)

DB2 trace type	DB2 trace class	DB2 IFCID	Record type
Performance	10	23	Start utility run, start of subtask.
		24	Utility change; phase info, subtask info
		25	Utility end
		219	LISTDEF
		220	Data set info
I/O Events			
Performance	4	6	Begin Read I/O, data set on DASD to buffer pool
		7	End Read I/O, data set on DASD to buffer pool
		8	Begin synchronous Write I/O
		9	Write end
		226	Page latch wait begin
		227	Page latch wait end
Lock Suspension Events			
Performance	6	44	Lock suspension or an identity call to the IRLM
		45	Lock resumption
		213	Beginning of a wait for a drain lock
		214	End of a wait for a drain lock
Performance	17	213	Beginning of a wait for a drain lock
		214	End of a wait for a drain lock
		215	Begin of a wait for the claim count to go to zero
		216	End of a wait for the claim count to go to zero
Page and Row Locking Events			
Performance	6	20	Page and row locking summary
		218	Lock avoidance summary
Exit Events			
Performance	13	11	Validation exit
		12	Encode edit exit
		19	Decode edit exit

Collecting Utility Activity data with a DB2 trace

To create an Utility Activity report, you must collect data by using a DB2 trace.

The easiest way to collect data for any OMEGAMON XE for DB2 PE report is to use the activate traces facility of the workstation monitor. You can use this facility to configure a DB2 trace and start and stop the trace from there. Using the activate

traces facility also ensures that you include all data in the DB2 trace that is needed for a report. For more information, see *Monitoring Performance from Performance Expert Client*.

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE(A) DEST(OPX) CLASS(1, 2, 3, 5, 7, 8) BUFSIZE(512)
-START TRACE(P) DEST(SMF) CLASS(30) IFCID(23,24,25,108.109.22.63)
  BUFSIZE(512)
-START TRACE(AU) DEST(OPX) CLASS(8) BUFSIZE(512)
```

Creating a Utility Activity trace

Use your ISPF editor to create a JCL job that creates a Utility Activity trace.

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

```
EDIT ---- ANDREW.OMPE.JOBS.UTTRA----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce an Utility Activity Report */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 //          MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 //          EXEC PGM=FPECMAN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.UTI,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.UTTRAC,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 UTILITY
0011 TRACE
0012 TYPE(UTILITY)
0013 ORDER (PRIMAUTH-PLANNAME-INSTANCE)
0014 EXEC
0015 ***** ***** 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 35. Utility Activity job stream using ISPF/PDF editor

Example of a short Utility Activity trace

The following example shows a trace produced from this JCL job.

LOCATION: PMODA11GANZLANGE	OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)	PAGE: 1-1
GROUP: N/P	UTILITY ACTIVITY TRACE	REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P		TO: NOT SPECIFIED
SUBSYSTEM: DA11		ACTUAL FROM: 07/25/13 10:14:19.95
DB2 VERSION: V10		PAGE DATE: 07/25/13

WORKLOAD(NONE)									
IDENTIFIED BY PRIMAUTH/PLANNAME/INSTANCE									
PRIMAUTH PLANNAME INSTANCE	START TIME SUBTASKS	ELAPSED TIME CPU TIME	ACTIVITY TYPE	UTILITY-ID DATABASE.PAGESET	JOB NAME STEP	SHRLEVEL TOT. ZIIP			
	SORT: DF DB2	DATA INDEX	OTHER		SORT CPU	SORT ZIIP			

MTS DSNUTIL X'CBB6392D08E1'	10:14:19.95 0 1 0	3.360279 0.651676	UTILITY LOAD	LDTAB1V9 DBPARALL.TSPARALL DBPARALL.XTAB1 DBPARALL.XTAB2	MTSLDA11 LOADTAB1	REFERENCE 0.077772			
	Y N	0 0	1		0.306264	0.000000			
MTS DSNUTIL X'CBB639305E39'	10:14:23.32 N/A N/A N/A	1.918446 0.323208	UTILITY RUNSTATS	LDTAB1V9 DBPARALL.TSPARALL DBPARALL.XTAB1 DBPARALL.XTAB2	MTSLDA11 LOADTAB1	REFERENCE 0.000000			
	N N	N/A N/A	N/A		N/A	N/A			
UTILITY TRACE COMPLETE									

The following sample Utility Activity trace is generated with the following command:

```
TRACE
TYPE(BIND)
ORDER (PRIMAUTH-PLANNAME-INSTANCE)
```

LOCATION: OMPDB51
GROUP: N/P
MEMBER: N/P
SUBSYSTEM: DB51
DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
UTILITY ACTIVITY TRACE

PAGE: 1-1
REQUESTED FROM: NOT SPECIFIED
TO: NOT SPECIFIED
ACTUAL FROM: 06/07/13 09:14:49.15
PAGE DATE: 06/07/13

WORKLOAD(NONE)
IDENTIFIED BY PRIMAUTH/PLANNAME/INSTANCE

PRIMAUTH PLANNAME INSTANCE	START TIME	ELAPSED TIME CPU TIME	ACTIVITY TYPE	OBJECT(S)
KOZS DSNBIND X'CB79D25996E1'	09:14:49.15	0.001552 0.000000	BIND BIND PLAN	PLANNAME: DSNTIA11
KOZS DSNBIND X'CB79D25F57AF'	09:14:55.12	0.000643 0.000000	BIND BIND PLAN	PLANNAME: DSNTIAD
KOZS DSNBIND X'CB79D266C962'	09:15:02.92	0.000280 0.000000	BIND BIND PLAN	PLANNAME: DSNREXX
KOZS DSNBIND X'CB79D2F497B1'	09:17:31.62	0.005636 0.000000	BIND FREE PLAN	PLANNAME: LGNADML
KOZS DSNBIND X'CB79D3411068'	09:18:51.94	0.027328 0.000000	BIND RBND PACK	LOCN: OMPDB51 COLL: UTRUN PKID: UTILV110 VRID: N/P CONS: X'195D036002523DD6'

UTILITY TRACE COMPLETE

Chapter 14. Record Trace reports

Record Trace reports show IFCID information as presented by DB2.

This IFCID information is used to produce other OMEGAMON XE for DB2 PE reports. The difference is that for other reports, this information can be interpreted, manipulated, or not included.

- A *summary* Record Trace report lists all selected records, together with a description, but without any record data. You can use this listing to determine what events occurred during the DB2 trace.
- A *short* Record Trace report presents nonserviceability data from selected records that appear on other OMEGAMON XE for DB2 PE reports. Some large IFCIDs (for example, system statistics) are presented similar to summary Record traces, without record data.
- A *long* Record Trace report presents serviceability and nonserviceability data from selected records.
- A *dump* Record Trace report presents selected records in the standard hexadecimal dump format. You can use the dump Record Trace report to view data that might contain unprintable characters.

“When to use Record Trace reports”

Use Record Trace reports if the information in other reports do not provide the required level of detail.

“Input for Record Trace reports”

Record Trace reports use IFCIDs from several DB2 trace types and trace classes.

“Collecting Record Trace data with a DB2 trace” on page 119

To create a Record Trace report, you must collect data by using a DB2 trace.

“Creating a Record Trace report” on page 119

Use your ISPF editor to create a JCL job that creates a Record Trace report.

When to use Record Trace reports

Use Record Trace reports if the information in other reports do not provide the required level of detail.

Input for Record Trace reports

Record Trace reports use IFCIDs from several DB2 trace types and trace classes.

Input to the Record Trace report set consists of all types of DB2 instrumentation data. The DB2 trace types and classes used by Record Trace reports are in the following list:

- DB2 trace type - Statistics

Table 10. DB2 trace type - Statistics: IFCIDs used for Record Trace reports

DB2 trace class	DB2 IFCIDs used	Class description
1 (Default Class)	1, 2, 105, 106, 202, 225	Information about system services, database statistics, statistics for the DBM1 address space, and information about the system parameters that were in effect when the trace was started. This default class is also activated when you omit the CLASS keyword from the START TRACE command when you start the statistics trace.
2	152	Installation-defined statistics record.
3	172, 196, 250, 258, 261, 262, 313, 330, 337	Deadlock, lock escalation, group buffer pool, data set extension information, and indications of long-running uncommitted reads, and active log space shortages.
4	173, 191, 192, 193, 194, 195, 203, 204, 205, 206, 207, 208, 209, 210, 235, 236, 238, 267, 268	DB2 exceptional conditions.
5	230	DB2 data sharing statistics record.
6	225	Storage statistics for the DB2 subsystem.
7	365	DRDA location statistics.
8	199	Data set I/O statistics.
9	369	Aggregate CPU and wait time statistics by connection type.

- DB2 trace type - Accounting

Table 11. DB2 trace type - Accounting: IFCIDs used for Record Trace reports

DB2 trace class	DB2 IFCIDs used	Class description
1 (Default Class)	3, 106, 239	Standard accounting data. This default class is also activated when you omit the CLASS keyword from the START TRACE command when you start the accounting trace.
2	200, 232	Entry or exit from DB2 event signalling.
3	6, 7, 8, 9, 32, 33, 44, 45, 117, 118, 127, 128, 170, 171, 174, 175, 213, 214, 215, 216, 226, 227, 242, 243, 321, 322, 329, 378, 379, 382, 383	Elapsed wait time in DB2.
4	151	Installation-defined accounting record.
5	187	Time spent processing IFI requests.
7	200, 232, 240	Entry or exit from DB2 event signalling for package and DBRM accounting.

Table 11. DB2 trace type - Accounting: IFCIDs used for Record Trace reports (continued)

DB2 trace class	DB2 IFCIDs used	Class description
8	6, 7, 8, 9, 32, 33, 44, 45, 117, 118, 127, 128, 170, 171, 174, 175, 213, 214, 215, 216, 226, 227, 239, 241 242, 243, 321, 322, 378, 379, 382, 383	Wait time for a package.
10	239	Package detail.

- DB2 trace type - Audit

Table 12. DB2 trace type - Audit: IFCIDs used for Record Trace reports

DB2 trace class	DB2 IFCIDs used	Class description
1 (Default Class)	140	Access attempts denied due to inadequate authorization. This default class is also activated when you omit the CLASS keyword from the START TRACE command when you start the audit trace.
2	141	Explicit GRANT and REVOKE.
3	142	CREATE, ALTER, and DROP operations against audited tables.
4	143	First change of audited object.
5	144	First read of audited object.
6	145	Bind time information about SQL statements that involve audited objects.
7	55, 83, 87, 169, 319	Assignment or change of authorization ID.
8	23, 24, 25, 219, 220	Utilities.
9	146	Installation-defined audit record.
10	269, 270	Trusted context information.
11	361	Audit administrative authorities.

- DB2 trace type - Monitor

Table 13. DB2 trace type - Monitor: IFCIDs used for Record Trace reports

DB2 trace class	DB2 IFCIDs used	Class description
1 (Default Class)	200	Standard accounting data. This default class is also activated when you omit the CLASS keyword from the START TRACE command when you start the monitor trace.

Table 13. DB2 trace type - Monitor: IFCIDs used for Record Trace reports (continued)

DB2 trace class	DB2 IFCIDs used	Class description
2	232	Entry or exit from DB2 event signalling. The information can be obtained by issuing a READS request for IFCID 147 or 148. In addition, monitor trace class 2 information is available in IFCID 3 in the accounting record. Monitor class 2 is equivalent to accounting class 2 and results in equivalent overhead. Monitor class 2 times appear in IFCIDs 147, 148, and 3 if either monitor trace class 2 or accounting class 2 is active.
3	6, 7, 8, 9, 32, 33, 44, 45, 117, 118, 127, 128, 170, 171, 174, 175, 213, 214, 215, 216, 226, 227, 242, 243, 321, 322, 378, 379, 382, 383	DB2 wait time for I/O, locks; resource usage information. The information can be obtained by issuing a READS request for IFCID 147 or 148. In addition, monitor trace class 3 information is available in the accounting record, IFCID 3. As with monitor class 2, monitor class 3 overhead is equivalent to accounting class 3 overhead. When monitor trace class 3 is active, DB2 can calculate the duration of a class 3 event, such as when an agent is suspended due to an unavailable lock. Monitor class 3 times appear in IFCIDs 147, 148, and 3, if either monitor class 3 or accounting class 3 is active.
4	155	Installation-defined monitor record.
5	187	Time spent processing IFI requests.
6	185	Changes to tables created with DATA CAPTURE CHANGES.
7	200, 232, 240	Entry or exit from DB2 event signalling for package accounting. The data traces the amount of time an agent spent in DB2 to process each package. If monitor trace class 2 is active, activating class 7 has minimal performance impact. Class 7 enables the IFCID 239 to be externalized.
8	6, 7, 8, 9, 32, 33, 44, 45, 51, 52, 56, 57, 117, 118, 127, 128, 170, 171, 174, 175, 213, 214, 215, 216, 226, 227, 239, 241, 242, 243, 321, 322, 378, 379, 382, 383	Wait time for a package. If monitor trace class 3 is active, activating class 8 has minimal performance impact. Class 8 enables the IFCID 239 to be externalized.
9	124	Enables statement level accounting. Provides information about statement details in IFCID 148. Monitor records do not contain class 10, but it shows up in IFCID 3 in the accounting record.

Table 13. DB2 trace type - Monitor: IFCIDs used for Record Trace reports (continued)

DB2 trace class	DB2 IFCIDs used	Class description
10	239	Package detail for buffer manager, lock manager and SQL statistics. It contains the same information as accounting class 10. Monitor records do not include class 10, but it shows up in IFCID 3 in the accounting record. Information from class 10 is written in additional sections of IFCID 239. However, monitor class 7 or 8 must be activated for IFCID 239 to be written. One of the following trace must also be activated before the IFCID 239 records are written: • Accounting class 7 • Accounting class 8 • Monitoring class 7 • Monitoring class 8
29	316, 318, 400, 401	Controls the subsystem-wide collection of statistics for SQL statements. Monitor class 29 must be activated for IFCID 316 records to be written for dynamic SQL statements and IFCID 401 records to be written for static sql statements.

- DB2 trace type - Performance

Table 14. DB2 trace type - Performance: IFCIDs used for Record Trace reports

DB2 trace class	DB2 IFCIDs used	Class description
1 (Default Class)	1, 2, 31, 42, 43, 76, 77, 78, 79, 102, 103, 105, 106, 107, 153	Background events. This default class is also activated when you omit the CLASS keyword from the START TRACE command when you start the performance trace.
2 (Default Class)	3, 68, 69, 70, 71, 72, 73, 74, 75, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 106, 174, 175	Subsystem events. This default class is also activated when you omit the CLASS keyword from the START TRACE command when you start the performance trace.
3 (Default Class)	22, 53, 55, 58, 59, 60, 61, 62, 63, 64, 65, 66, 92, 95, 96, 97, 106, 112, 173, 177, 233, 237, 250, 272, 273, 325	SQL events. This default class is also activated when you omit the CLASS keyword from the START TRACE command when you start the performance trace.
4	6, 7, 8, 9, 10, 29, 30, 105, 106, 107, 127, 128, 226, 227, 321, 322	Reads to and writes from the buffer and EDM pools.
5	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 104, 106, 114, 115, 116, 117, 118, 119, 120, 228, 229	Write to log; archive log.
6	20, 44, 45, 105, 106, 107, 172, 196, 213, 214, 218, 337	Summary lock information.

Table 14. DB2 trace type - Performance: IFCIDs used for Record Trace reports (continued)

DB2 trace class	DB2 IFCIDs used	Class description
7	21, 105, 106, 107, 223	Detailed lock information.
8	13, 14, 15, 16, 17, 18, 105, 106, 107, 125, 221, 222, 231, 305, 311, 363	Data scanning detail.
9	26, 27, 28, 95, 96, 106	Sort detail.
10	23, 24, 25, 90, 91, 105, 106, 107, 108, 109, 110, 111, 201, 256	BIND, commands, and utilities detail.
11	46, 47, 48, 49, 50, 51, 52, 56, 57, 93, 94, 106, 113	Execution unit switch and latch contentions.
12	98, 99, 100, 101, 106	Storage manager.
13	11, 12, 19, 105, 106, 107	Edit and validation exits.
14	67, 106, 121, 122	Entry from and exit to an application.
15	154	Installation-defined performance record.
16	157, 158, 159, 160, 161, 162, 163, 167, 183	Distributed processing.
17	211, 212, 213, 214, 215, 216	Claim and drain information.
18	197	Event-based console messages.
19	370, 371	Data set open and close activity.
20	249, 250, 251, 256, 257, 261, 262, 267, 268	Data sharing coherency summary.
21	255, 259, 263	Data sharing coherency detail.
22	314	Authorization exit parameters.
23	327	Language environment runtime diagnostics.
24	380, 499	Stored procedure detail.
29	-	Reserved! (not used anymore)

- DB2 trace type - Global

Table 15. DB2 trace type - Global: IFCIDs used for Record Trace reports

DB2 trace class	DB2 IFCIDs used
1 (Default Class)	106, (132, 134, 138)
2	106, (131, 133, 139)
3	0, 38, 46, 47, 48, 49, 50, 51, 52, 56, 57, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 93, 94, 106, 114, 115, 116, 117, 174, 175, 228, 229, 252, 260, 265, 266, 267, 268

Table 15. DB2 trace type - Global: IFCIDs used for Record Trace reports (continued)

DB2 trace class	DB2 IFCIDs used
4	106, (130)
5	190, 249 (135, 136, 137,) (247, 248)
6	156
7	164, 165, 166
8	168
9	180, 181, 182

Collecting Record Trace data with a DB2 trace

To create a Record Trace report, you must collect data by using a DB2 trace.

A Record trace can be very long. Generally, run a Record trace on one component at a time to focus on a problem. The following example collects records associated with Accounting.

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

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE(A) DEST(OPX) CLASS(1, 2, 3, 5, 7, 8) BUFSIZE(512)
-START TRACE(P) DEST(OPX) CLASS(17,16,14,13,12,11,10,9,8,7,4,2,22,21,20,30)
IFCID(20,22,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,53,55,58,59,60,61,
62,63,64,65,66,76,77,78,79,80,81,92,97,102,103,104,112,114,115,116,117,118,
119,120,124,129,140,141,142,143,144,145,146,147,149,150,151,152,153,154,
155,169,170,171,177,178,179,186,188,191,192,193,194,195,198,203,204,205,206,
207,208,209,210,218,219,220,224,228,229,233,235,236,237,238,239,243,247,254,
258,272,273,312,313,324,325,329,330,331,332,333,334) BUFSIZE(512)
-START TRACE(S) DEST(OPX) CLASS(6,5,3,1) BUFSIZE(512)
```

Creating a Record Trace report

Use your ISPF editor to create a JCL job that creates a Record Trace report.

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

```

EDIT ---- ANDREW.OMPE.JOBS.RECTRC----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce an SQL activity trace */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 //          MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 //          EXEC PGM=FPECMAIN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.TRACE2,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.REC01,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 RECTRACE
0011 TRACE
0012 LEVEL(LONG)
0013 EXEC
***** ***** BOTTOM OF DATA *****
F1=Help      F2=Split    F3=Exit      F5=Rfind    F6=Rchange  F7=Up
F8=Down      F9=Swap     F10=Left   F11=Right  F12=Cancel

```

Figure 36. Record trace job stream using ISPF/PDF editor

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

Chapter 15. Housekeeping reports

These reports are not strictly performance reports. They present information about the DB2 system and user activity.

They provide valuable information, when used together with Accounting and Statistics reports, and help to gain an overview of the DB2 system.

“The Audit report set”

With DB2 Audit data you can track DB2 resource access. Audit reports and traces show information about the user of an auditable object and the time and type of action performed on the object.

“The System Parameters report” on page 124

This report provides information about the configuration of the DB2 system being monitored.

The Audit report set

With DB2 Audit data you can track DB2 resource access. Audit reports and traces show information about the user of an auditable object and the time and type of action performed on the object.

Audit reports group events in the order of event type, ORDER identifier, and timestamp.

Audit traces show events listed chronologically.

You can use Audit reports and traces to review Audit data by OMEGAMON XE for DB2 PE identifiers and choose between a wide range of options for presenting DB2 Audit data.

You can order Audit reports and traces by the object of the Audit event (for example, authorization failures per table).

You can produce member-scope and group-scope Audit reports and traces.

- Member-scope reporting presents data member by member.
- Group-scope reporting merges the data that is produced by each group member to give a full picture of the use of a shared resource.

You can produce Audit reports and traces of individual Audit events in the order in which they occurred, or reports of aggregated Audit data.

You can specify the level of detail in Audit reports and traces by specifying any combination of the following event types:

- Authorization failures
- Changes to authorization identifiers
- DDL operations against auditable tables
- DML statements at bind of auditable tables
- Grants or revokes of privileges
- Read/write access to auditable tables
- Utility access to auditable tables

“When to use Audit reports”

Use Audit reports or traces regularly to assist in your database administration and as part of your regular monitoring policy. More specifically, use the Audit reports if Accounting reports show unexpected numbers of authorization failures.

“Collecting Audit data with a DB2 trace”

To create an Audit report, you must collect data by using a DB2 trace.

“Creating an Audit trace”

Use your ISPF editor to create a JCL job that creates an Audit trace.

When to use Audit reports

Use Audit reports or traces regularly to assist in your database administration and as part of your regular monitoring policy. More specifically, use the Audit reports if Accounting reports show unexpected numbers of authorization failures.

You can use Audit reports to monitor:

Usage of sensitive data

Tables that contain sensitive data, such as employee salary records, should probably be defined with AUDIT ALL. You can report usage by table and by authorization ID to look for access by unusual IDs, at unusual times, or of unexpected types. You also want to record any ALTER or DROP operations that affect the data.

Grants of critical privileges

Authorities such as SYSADM and DBADM and explicit privileges over sensitive data, such as an Update privilege on records of accounts payable, must be monitored carefully. A query of the DB2 catalog can show who holds such a privilege at a particular time. The Audit records can reveal whether the privilege was granted and then revoked in a period of time.

Unsuccessful access attempts

Some unsuccessful access attempts are only user errors, but others can be attempts to violate security. All must be investigated. If you have sensitive data, always use Audit class 1 trace data.

Collecting Audit data with a DB2 trace

To create an Audit report, you must collect data by using a DB2 trace.

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

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE(AU) DEST(OPX) CLASS(1,2,3,4,5,6) BUFSIZE(512)
-START TRACE(P) DEST(OPX) CLASS(30) IFCID(24,55,83,87,105,107,169) BUFSIZE(512)
```

Creating an Audit trace

Use your ISPF editor to create a JCL job that creates an Audit trace.

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

```

EDIT ---- ANDREW.OMPE.JOBS.AUDTRC----- Columns 001 072
Command ==> _____ Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce an Audit trace */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.AUDIT,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.AUDT1,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 AUDIT
0011 TRACE
0012 SCOPE (MEMBER)
0013 EXEC
***** ***** BOTTOM OF DATA *****
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 37. Audit job stream using ISPF/PDF editor

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

Example of a member-scope Audit trace

The following example shows the resulting member-scope Audit trace.

LOCATION: LOCATI_2	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 1-1
GROUP: GROUP_02	AUDIT TRACE	REQUESTED FROM: NOT SPECIFIED
MEMBER: MEMBER_2		TO: NOT SPECIFIED
SUBSYSTEM: SYS2		ACTUAL FROM: 07/17/13 04:21:44.17
DB2 VERSION: V10	SCOPE: MEMBER	PAGE DATE: 07/17/13

PRIMAUTH	CORRNAME	CONNTYPE	ORIGAUTH	CORRNMBR	INSTANCE	PLANNAME	CONNECT	TIMESTAMP	TYPE	DETAIL
AUTH_20	CORNME30	TSO	ORAUTH30	'BLANK'	A4F4970F2B37	PLAN_20	CONN_20	04:21:44.26	DDL	TABLE NAME: NHEMP DATABASE : DBASE1 TEXT: ALTER TABLE N HEMP AUDIT ALL
AUTH_20	CORNME30	TSO	ORAUTH30	'BLANK'	A4F497ADC393	PLAN_20	CONN_20	04:23:27.14	DML	TYPE : 1ST READ DATABASE: DBASE1 PAGESET : PSET1 TABLE OBID: 7 LOG RBA : X'00000EA2CAF7'
AUTH_20	CORNME30	TSO	ORAUTH30	'BLANK'	A4F497ADC393	PLAN_20	CONN_20	04:23:27.34	UTILITY	NAME : LOAD DATABASE: DBASE1 PAGESET : PSET1 ID : DSNTX PHASE: RELOAD TYPE : RECORD
AUTH_20	CORNME30	TSO	ORAUTH30	'BLANK'	A4F497ADC393	PLAN_20	CONN_20	04:23:28.88	DML	TYPE : 1ST WRITE DATABASE: DBASE1 PAGESET : PSET1 TABLE OBID: 7 LOG RBA : X'00000EA2D200'
AUTH_20	CORNME30	TSO	ORAUTH30	'BLANK'	A4F497D49C07	PLAN_20	CONN_20	04:24:04.97	AUTHFAIL	AUTHID CHECKED: XXASP07 OBJECT TYPE : TABLE SOURCE OBJECT : SYSDATABASE TARGET OBJECT : SYSDATABASE TEXT: REVOKE SYSADM FROM XXASP07 BY ALL PRIVILEGE: SELECT REASON: 8 RC: 4 SOURCE OWNER: SYSIBM TARGET OWNER: SYSIBM
AUTH_20	CORNME30	TSO	ORAUTH30	'BLANK'	A4F497D49C07	PLAN_20	CONN_20	04:24:06.30	AUTHCNTL	REVOKER: XXASP09 REASON: N/A OBJECT TYPE: TSPACE TEXT: REVOKE SYSADM FROM XXASP07 BY ALL SQLCODE: 0
AUTH_20	CORNME30	TSO	ORAUTH30	'BLANK'	A4F49D0AA847	PLAN_20	CONN_20	04:29:20.37	AUTHCHG	TYPE: KERBEROS COMMUNICATION ADDRESS TYPE: SNA DERIVED LOCAL USERID : LOCALAD PRINCIPAL NAME : /.../db2dev/mckueogh REQ COMMUNICATION ADDR: FA0016354B98382
AUTH_20	CORNME30	DRDA	ORAUTH30	CN30	A4F49B66B12D	PLAN_20	CONN_20	04:40:14.57	BIND	PACKAGE: DSNAPC3.DSNESPCS.DSNESM68.X'148C2637049FB9B0' TYPE: SEL-QUERY STMT# 71 ISOLATION(CS) TEXT: SELECT * FROM XXASP09.NHDEPT DATABASE: DBASE1 TABLE OBID: 7
AUTH_20	CORNME30	DRDA						04:42:49.13	BIND	PACKAGE: DSNAPC3.DSNESPCS.DSNESM68.X'148C2637049FB9B0'

ORAUTH30 CN30 A4F49B66B12F
PLAN_20 CONN_20
REQLOC: LOCATI_30
AUDIT TRACE COMPLETE

TYPE: SEL-QUERY STMT# 71 ISOLATION(CS)
TEXT: SELECT * FROM XXASP09.NHDEPT
DATABASE: DBASE1 TABLE OBID: 7

The System Parameters report

This report provides information about the configuration of the DB2 system being monitored.

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

Some parameters, such as buffer pool and group buffer pool attributes can be changed while a system is active. If the appropriate DB2 trace class is active, the changes are recorded in the System Parameters report in the order of occurrence.

“When to use System Parameters reports”

Use the System Parameters report regularly as part of your system monitoring policy.

“Collecting System Parameters data with a DB2 trace”

To create a System Parameters report, you must collect data by using a DB2 trace.

“Creating a System Parameters report”

Use your ISPF editor to create a JCL job that creates a System Parameters report.

When to use System Parameters reports

Use the System Parameters report regularly as part of your system monitoring policy.

You can also use this report together with a Statistics report to determine whether you can improve a system's performance by altering its system settings.

Collecting System Parameters data with a DB2 trace

To create a System Parameters report, you must collect data by using a DB2 trace.

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

Alternatively, you can start a DB2 trace as follows:

```
-START TRACE(S) DEST(OPX) CLASS(5) BUFSIZE(512)  
-START TRACE(P) DEST(OPX) CLASS(30) IFCID(106,201,202,256) BUFSIZE(512)
```

Creating a System Parameters report

Use your ISPF editor to create a JCL job that creates a System Parameters report.

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


```

EDIT ---- ANDREW.OMPE.JOBS.SYPTRC----- Columns 001 072
Command ==> Scroll ==> CSR_
***** ***** TOP OF DATA *****
0001 /* JCL to produce a System Parameters report */
0002 //AWOMPEA JOB (TTS1,YUS7),'ANDREW',
0003 // MSGCLASS=V,CLASS=D,NOTIFY=ANDREW
0004 // EXEC PGM=FPECMAN
0005 //STEPLIB DD DSN=OMPE.RKANMOD,DISP=SHR
0006 //INPUTDD DD DSN=ANDREW.OMPE.TRACES.SYSP,DISP=SHR
0007 //DPMOUTDD DD DSN=ANDREW.OMPE.OUT.SYSPT1,DISP=OVR
0008 //JOBSUMDD DD SYSOUT=A
0009 //SYSIN DD *
0010 SYSPARMS
0011 EXEC
0012 ***** ***** BOTTOM OF DATA *****
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 38. System parameters job stream using ISPF/PDF editor

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

Example of the System Parameters report

The following example shows the resulting System Parameters report.

```

LOCATION: PMODBE1                      OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)      PAGE: 1-1
GROUP: DBE1
MEMBER: SE11
SUBSYSTEM: SE11
DB2 VERSION: V11
ACTUAL FROM: 07/15/13 03:46:00.00

MVS PARMLIB UPDATE PARAMETERS (DSNTIPM)
-----
SUBSYSTEM DEFAULT (SSID).....SE11
SUPPRESS SOFT ERRORS (SUPERRS).....YES

STORAGE SIZES INSTALLATION PARMS (DSNTIPC,DSNTIPE,DSNTIPE1)
-----
MAX NO OF DATA SETS CONCURRENTLY IN USE (DSMAX).....20,000
EDM STATEMENT CACHE SIZE IN KB (EDMSTMT).....113,386
EDM DBD CACHE SIZE IN KB (EDMDBDC).....102,400
EDM SKELETON POOL SIZE IN KB (EDM_SKELETON_POOL).....102,400
MAXIMUM SIZE OF EDM POOL IN BYTES (EDMPOOL).....400,000
MAXIMUM SIZE OF SORT POOL IN BYTES (SRTPPOOL).....10,240,000
MAX IN-MEMORY SORT SIZE (MAXSORT_IN_MEMORY).....1,000
MAXIMUM SIZE OF RID POOL IN KB (MAXRBLK).....400,000
MAX NO OF USERS CONCURRENTLY RUNNING IN DB2 (CTHREAD).....400
MAX NO OF CONCURRENT REMOTE ACTIVE CONNECTIONS (MAXDBAT).....200
MAX NO OF REMOTE CONNECTIONS (CONDBAT).....10,000
MAX NO OF TSO CONNECTIONS (IDFORE).....200
MAX NO OF BATCH CONNECTIONS (IDBACK).....200
MAXIMUM KEPT DYNAMIC STATEMENTS (MAXKEEPD).....5,000
MAX OPEN FILE REFS (MAXOFILR).....100
CONTRACT THREAD STORAGE (CONSTOR).....YES
MANAGE THREAD STORAGE (MINSTOR).....NO
LONG-RUNNING READER IN MINUTES (LDRTHLD).....10
DDL TIMEOUT FACTOR (DDLTOX).....1
INDEX CLEANUP THREADS (INDEX_CLEANUP_THREADS).....10

TRACING, CHECKPOINT & PSEUDO-CLOSE PARAMETERS (DSNTIPN)
-----
START AUDIT TRACE (AUDITST).....NO
START GLOBAL TRACE (TRACSTR).....NO
TRACE TABLE SIZE IN 4K BYTES (TRACETBL).....16
START SMF ACCOUNTING (SMFACT).....YES
START SMF STATISTICS (SMFSTAT).....1,3,4,5,6
STATISTICS TIME INTERVAL IN MINUTES (STATIME).....1
SYNCHRONIZATION INTERVAL WITHIN THE HOUR (SYNVAL).....NO
ONLINE DATASET STATISTICS TIME INTERVAL IN MIN. (DSSTIME).....5
START MONITOR TRACE (MON).....1
MONITOR BUFFER SIZE IN BYTES (MONSIZE).....1,048,576
UNICODE IFCIDS (UIFCIDS).....YES
DDF/RRSAF ACCUM (ACCUMACC).....10
AGGREGATION FIELDS (ACCUMUID).....0
COMPRESS SMF RECS (SMFCOMP).....OFF

IRLM INSTALLATION PARAMETERS (DSNTIPI)
-----
IRLM SUBSYSTEM NAME (IRLMSID).....IE11
IRLM RESOURCE TIMEOUT IN SECONDS (IRLMRWT).....30
IRLM AUTOMATIC START (IRLMAUT).....YES
IRLM START PROCEDURE NAME (IRLMPRC).....SE11IRLM
SECONDS DB2 WILL WAIT FOR IRLM START (IRLMSWT).....120
U LOCK FOR REPEATABLE READ OR READ STABILITY (RRULOCK).....YES
X LOCK FOR SEARCHED UPDATE/DELETE (XLKUPDLT).....NO
IMS/BMP TIMEOUT FACTOR (BMPTOUT).....4
IMS/DLI TIMEOUT FACTOR (DLITOUT).....6
WAIT FOR RETAINED LOCKS (RETLWAIT).....0
ENABLE DB CHECKING.....NO

IRLM PROCESSING PARAMETERS
-----
WAIT TIME FOR LOCAL DEADLOCK.....5,000
NUMBER OF LOCAL CYCLES PER GLOBAL CYCLE.....1
TIMEOUT INTERVAL.....30
IRLM MAXIMUM CSA USAGE ALLOWED.....0
Z/OS LOCK TABLE HASH ENTRIES.....1,048,576
PENDING NUMBER OF HASH ENTRIES.....0
Z/OS LOCK TABLE LIST ENTRIES.....8,282
MAX 31-BIT IRLM PRIVATE STORAGE.....0
MAX 64-BIT IRLM PRIVATE STORAGE.....0

ARCHIVE LOG INSTALLATION PARAMETERS (DSNTIPA)
-----
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).....100
SECONDARY SPACE ALLOCATION (SECQTY).....10
ARCHIVE LOG BLOCK SIZE IN BYTES (BLKSIZE).....24,576
MAXIMUM READ TAPE UNITS (MAXRTU).....2
TAPE UNIT DEALLOCATION PERIOD (DEALLCT).....0000:00
MAX NUMBER OF DATASETS RECORDED IN BSDS (MAXARCH).....10,000
FIRST ARCHIVE COPY MASS STG GROUP NAME.....'NONE'
SECOND ARCHIVE COPY MASS STG GROUP NAME.....'NONE'
DAYS TO RETAIN ARCHIVE LOG DATA SETS (ARCRTN).....30
ISSUE WTOR BEFORE MOUNT FOR ARCHIVE VOLUME (ARCWTOR).....YES
COMPACT DATA (COMPACT).....NO
QUIESCE PERIOD (QUIESCE).....5
SINGLE VOLUME (SVOLARC).....NO

```

LOCATION: PMODBE1
GROUP: DBE1
MEMBER: SE11
SUBSYSTEM: SE11
DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
SYSTEM PARAMETERS REPORT

PAGE: 1-2

ACTUAL FROM: 07/15/13 03:46:00.00

DISTRIBUTED DATA FACILITY PANEL 2 (DSNTIP5)

TCP/IP ALREADY VERIFIED (TCPALVER).....NO
EXTRA BLOCKS REQ (EXTRAREQ).....100
EXTRA BLOCKS SRV (EXTRASRV).....100
TCP/IP KEEPALIVE (TCPKPALV).....120
CONNECTION QUEUE MAX DEPTH (MAXCONQN).....0
CONNECTION QUEUE MAX WAIT (MAXCONQW).....0
POOL THREAD TIMEOUT (POOLINAC).....120

PROTECTION INSTALLATION PARAMETERS (DSNTIPP)

ARCHIVE LOG RACF PROTECTION (PROTECT).....NO
DB2 AUTHORIZATION ENABLED (AUTH).....YES
PLAN AUTHORIZATION CACHE SIZE (AUTHCACH).....3,072
PACKAGE AUTHORIZATION CACHE SIZE (CACHEPAC).....5,242,880
ROUTINE AUTHORIZATION CACHE SIZE (CACHERAC).....5,242,880
AUTH EXIT CHECK (AUTHEXIT CHECK).....PRIMARY
AUTH EXIT CACHE REFRESH (AUTHEXIT CACHEREFRESH).....NONE
SYSTEM ADMINISTRATOR 1 AUTHORIZATION ID (SYSADM).....HELM
SYSTEM ADMINISTRATOR 2 AUTHORIZATION ID (SYSADM2).....SYSADM
SYSTEM OPERATOR 1 AUTHORIZATION ID (SYSOPR1).....HELM
SYSTEM OPERATOR 2 AUTHORIZATION ID (SYSOPR2).....EMIL
DEFAULT (UNKNOWN) USER AUTHORIZATION ID (DEFLTID).....IBMUSER
RESOURCE LIMIT TABLE CREATOR AUTH ID (RLFAUTH).....SYSIBM
BIND NEW PACKAGE (BINDNV).....BINDADD
DBA CREATE VIEW (DBACRVW).....NO

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'

DB2 VERSION INSTALL (DSNTIPA1)

DATA SHARING ENABLED (DSHARE).....YES
INSTALL TYPE (NEWFUN).....YES

DEFINE GROUP OR MEMBER (DSNTIPK)

GROUP NAME (GRPNAME).....DBE1
MEMBER NAME (MEMBNAME).....SE11
DEL CF STRUCTS (DEL_CFSTRUCTS_ON_RESTART).....NO

DISTRIBUTED DATA FACILITY PANEL 1 (DSNTIPR)

DDF STARTUP OPTION (DDF).....AUTO
RLST ACCESS ERROR (RLFERRD).....NOLIMIT
RESYNCHRONIZATION INTERVAL IN MINUTES (RESYNC).....2
DBAT STATUS (CMTSTAT).....INACTIVE
IDLE THREAD TIMEOUT INTERVAL (IDTHTOIN).....120
EXTENDED SECURITY (EXTSEC).....YES
MAX TYPE 1 INACTIVE THREADS (MAXTYPE1).....0

LOCK ESCALATION PARAMETERS (DSNTIPJ)

MAX PAGE OR ROW LOCKS PER TABLE SPACE (NUMLKTS).....2,000
MAX PAGE OR ROW LOCKS PER USER (NUMLKUS).....10,000

LOG INSTALLATION PARAMETERS (DSNTIPL,DSNTIPH)

OUTPUT BUFFER SIZE IN K BYTES (OUTBUFF).....4,000
DBM1 STORAGE FOR FAST LOG (LOGAPSTG).....N/A
CHECKPOINT TYPE (CHKTYPE).....SINGLE
RECORDS/CHECKPOINT (CHKLOGR).....N/P
MINUTES/CHECKPOINT (CHKMINS).....N/P
CHECKPOINT FREQUENCY (CHKFREQ).....500,000
UR CHECK FREQUENCY (URCHKTH).....0
UR LOG RECORD WRITTEN THRESHOLD IN KB (URLGWTH).....0
LIMIT BACKOUT (LBACKOUT).....AUTO
BACKOUT DURATION (BACKODUR).....5
PSEUDO-CLOSE FREQUENCY (PCLOSEN).....10
PSEUDO-CLOSE TIMER (PCLOSET).....10
CHECKPOINTS BETWEEN LEVEL ID UPDATES (DLDFREQ).....5
NUMBER OF ACTIVE LOG COPIES (TWOACTV).....1
NUMBER OF ARCHIVE LOG COPIES (TWOARCH).....1
COPY 1 PREFIX (ARCPFX1).....DBE1.SE11.ARCHLOG1
COPY 2 PREFIX (ARCPFX2).....DBE1.SE11.ARCHLOG2
TIMESTAMP ARCHIVE LOG DATA SETS (TSTAMP).....YES

LOCATION: PMODBE1
GROUP: DBE1
MEMBER: SE11
SUBSYSTEM: SE11
DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
SYSTEM PARAMETERS REPORT

PAGE: 1-3

ACTUAL FROM: 07/15/13 03:46:00.00

APPLICATION PROGRAMMING DEFAULTS PANEL 1 (DSNTIPF)

DEFAULT HOST LANGUAGE (DEFLANG).....IBMCOB
DECIMAL POINT OPTION (DECIMAL).....PERIOD
DEFAULT DELIMITER (DELIM).....DEFAULT
DEFAULT SQL DELIMITER (SQLDELI).....DEFAULT
DIST SQL STRING DELIMITER (DSQLDELI).....APOST
DEFAULT MIXED GRAPHIC (MIXED).....NO
EBCDIC SBCS CCSID (SCCSID).....1,148
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
UNICODE SBCS CCSID (USCCSID).....367
UNICODE MBCS CCSID (UMCCSID).....1,208
UNICODE GBCS CCSID (UGCCSID).....1,200
DEFAULT ENCODING SCHEME (ENSCHME).....EBCDIC
APPLICATION ENCODING (APPENSCH).....EBCDIC
LOCALE LC TYPE (LC CTYPE).....'BLANK'
DECFLOAT ROUND MODE (DEF_DECFLOAT_ROUND_MODE)...ROUND_HALF_EVEN

SQL OBJECT DEFAULTS PANEL (DSNTIP7,DSNTIP71)

OBJECT CREATE FORMAT (OBJECT_CREATE_FORMAT).....EXTENDED
UTILITY OBJECT CONVERSION (UTILITY_OBJECT_CONVERSION).....NONE
VARY DS CONTROL INTERVAL (DSVCI).....YES
TABLE SPACE ALLOCATION IN KB (TSQTY).....0
INDEX SPACE ALLOCATION IN KB (IXQTY).....0
OPTIMIZE EXTENT SIZING (MGEXTSZ).....NO
PAD INDEX BY DEFAULT (PADIX).....NO
DEFAULT PARTITION SEGSIZE (DPSEGSZ).....32
PERCENT FREE FOR UPDATE (PCTFREE_UPD).....0
DEFINE DATA SETS (IMPDSEDEF).....YES
USE DATA COMPRESSION (IMPTSCMP).....NO
LIMIT KEY CONV PART TAB (IX_TB_PART_CONV_EXCLUDE).....NO
LOB INLINE LENGTH (LOB_INLINE_LENGTH).....0

PERFORMANCE AND OPTIMIZATION (DSNTIP8,DSNTIP81)

CACHE DYNAMIC SQL (CACHEDYN).....YES
OPTIMIZATION HINTS ALLOWED (OPTHINTS).....NO
EVALUATE UNCOMMITTED (EVALUNC).....NO
SKIP UNCOMM INSERTS (SKIPUNCI).....NO
IMMEDWRITE OVERRIDE FLAG (IMMEDWRI).....NO
REBIND PLANMGMT DEFAULT (PLANMGMT).....EXTENDED
PLANMGMTSCOPE DEFAULT (PLANMGMTSCOPE).....STATIC
PACKAGE RELEASE COMMIT (PKGREL_COMMIT).....YES
RANDOMIZE XML DOCID (XML_RANDOMIZE_DOCID).....NO
BLOCK OPT 1 ROW SORT (OPT1ROWBLOCKSORT).....NO
CURRENT DEGREE (CDSSRDEF).....1
MAX DEGREE OF PARALLELISM (PARAMDEG).....0

APPLICATION PROGRAMMING DEFAULTS PANEL 2 (DSNTIP4,DSNTIP41)

MINIMUM DIVIDE SCALE (DECDIV3).....NO
DECIMAL ARITHMETIC (DECARTH).....15
USE FOR DYNAMIC RULES (DYNRULS).....YES
STATIC DESCRIBE (DESCSTAT).....YES
DATE FORMAT (DATE).....ISO
TIME FORMAT (TIME).....ISO
LOCAL DATE LENGTH (DATELEN).....N/A
LOCAL TIME LENGTH (TIMELEN).....N/A
IMPLICIT TIMEZONE.....CURRENT
STD SQL LANGUAGE (STDSQL).....NO
PAD NULL-TERMINATED (PADNTSTR).....NO
APPL COMPAT LEVEL (APPLCOMPAT).....V11R1
LIKE BLANK INSIGNIFICANT (LIKE_BLANK_INSIGNIFICANT).....NO

OPERATOR FUNCTIONS INSTALLATION PARAMETERS (DSNTIPO)

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).....NO
SITE TYPE (SITETYP).....LOCALSITE
TRACKER SITE (TRKRSITE).....NO
READ COPY2 ARCHIVE (ARC2FRST).....NO
REAL TIME STATS (STATSINT).....30
STATISTICS FEEDBACK (STATFDBK_SCOPE).....ALL

ROUTINE PARAMETERS (DSNTIPX)

MAX ABEND COUNT (STORMXAB).....0
TIMEOUT VALUE (STORTIME).....180
WLM ENVIRONMENT (WLMENV).....WLMENV
MAX OPEN CURSORS (MAX_NUM_CUR).....500
MAX STORED PROCS (MAX_ST_PROC).....2,000

BUFFER POOL PARAMETERS (DSNTIP1)

DEFAULT 4-KB BUFFER POOL FOR USER DATA (TBSBP00L).....BP2
DEFAULT 8-KB BUFFER POOL FOR USER DATA (TBSBP8K).....BP8K0
DEFAULT 16-KB BUFFER POOL FOR USER DATA (TBSBP16K).....BP16K0
DEFAULT 32-KB BUFFER POOL FOR USER DATA (TBSBP32K).....BP32K
DEFAULT BUFFER POOL FOR USER INDEXES (IDXBP00L).....BP1

QUERY ACCELERATOR PREFERENCES (DSNTIP82)

ACCELERATOR STARTUP OPTION (ACCEL).....NO
GET ACCEL ARCHIVE (GET_ACCEL_ARCHIVE).....NO
ACCELERATION OPTIONS (QUERY_ACCEL_OPTIONS).....NONE

```

LOCATION: PMODBE1                      OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)      PAGE: 1-4
GROUP: DBE1                          SYSTEM PARAMETERS REPORT
MEMBER: SE11
SUBSYSTEM: SE11                      ACTUAL FROM: 07/15/13 03:46:00.00
DB2 VERSION: V11
MAX DEGREE FOR DPSI (PARAMDEG_DPSI).....0
STAR JOIN ENABLING (STARJOIN) .....DISABLE
MAX DATA CACHING IN MB (MXDTCACH).....20
CURRENT REFRESH AGE (REFSHAGE).....0
CURRENT MAINT TYPE (MAINTYPE).....SYSTEM

LOCATION: PMODBE1                      OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)      PAGE: 1-5
GROUP: DBE1                          SYSTEM PARAMETERS REPORT
MEMBER: SE11
SUBSYSTEM: SE11                      ACTUAL FROM: 07/15/13 03:46:00.00
DB2 VERSION: V11

OTHER SYSTEM PARAMETERS
-----
DUAL BSDS MODE (TWOBSDS).....YES
ROLL UP PARALLEL TASK ACCOUNTING (PTASKROL).....YES
NO. PAGES SMALL TABLE THRESHOLD (NPGTHRSH).....0
OFFLOAD OPTION (OFFLOAD).....YES
SU CONVERSION FACTOR.....281
MINIMUM DIVIDE SCALE (MINDVSCL).....NONE
STAR JOIN THRESHOLD (SJTABLES).....10
ONLINE SYSTEM PARM USER ID MONITOR.....N/P
ONLINE SYSTEM PARM CORREL ID MONITOR.....N/P
ONLINE SYSTEM PARM TIME CHANGED.....N/P
ONLINE SYSTEM PARM TYPE.....N/P
DB2-SUPPLIED DECP INDICATOR.....X'D5'
MAX CONCURRENT PKG OPS (MAX_CONCURRENT_PKG_OPS).....10
ADMIN SCHEDULER JCL PROC NAME (ADMTPROC).....N/P
FREE LOCAL CACHED STATEMENTS (CACHEDYN_FREELocal).....1
INDEX I/O PARALLELISM (INDEX_IO_PARALLELISM).....YES
ZOSMETRICS.....YES
USE TRACKMOD FOR IMPLICIT TS (IMPTKMOD).....YES
DSSIZE FOR IMPLICIT TS (IMPDSSIZE).....4
ENABLE MULTIPLE INDEX ACCESS (SUBQ_MIDX).....YES
SPT01 INLINE LENGTH (SPT01_INLINE_LENGTH).....32,138
DDF COMPATIBILITY (DDF_COMPATIBILITY).....NO
DYN STMT CACHE STOR (CACHE_DEP_TRACK_STOR_LIM).....0

DB2 UTILITIES PARAMETERS (DSNTIP6,DSNTIP61,DSNTIP62)
-----
TEMPORARY UNIT NAME (VOLTDEVT).....SYSDA
STATISTICS HISTORY (STATHIST).....NONE
STATISTICS ROLLUP (STATROLL).....YES
UTILITY TIMEOUT FACTOR (UTIMOUT).....6
TEMPLATE TIME (TEMPLATE_TIME).....UTC
MAXIMUM DEGREE OF UTILITY PARALLELISM (PARAMDEG_UTIL).....99
SYSTEM-LEVEL BACKUPS (SYSTEM_LEVEL_BACKUPS).....NO
RESTORE/RECOVER (RESTORE_RECOVER_FROMDUMP).....NO
DUMP CLASS NAME (UTILS_DUMP_CLASS_NAME).....'BLANK'
MAXIMUM TAPE UNITS (RESTORE_TAPEUNITS).....0
DEFAULT TEMPLATE (FCCOPYDDN).....
  DBE1.&DB..&SN..N&DSNUM..&U
REORG PART SORT NPSI (REORG_PART_SORT_NPSI).....AUTO
REORG LIST PROCESSING (REORG_LIST_PROCESSING).....PARALLEL
REORG MAPPING DATABASE (REORG_MAPPING_DATABASE).....N/P
REORG DROP PBG PARTS (REORG_DROP_PBG_PARTS).....NO
REORG IGNORE FREE SPACE (REORG_IGNORE_FREESPACE).....NO

CURRENT QUERY ACCEL (QUERY_ACCELERATION).....NONE

WORKFILE DATABASE PANEL (DSNTIP91)
-----
MAX TEMP STORAGE PER AGENT IN MB (MAXTEMPS).....0
MAX TEMP RID (MAXTEMPS_RID).....NOLIMIT
AGENT LEVEL THRESHOLD (WFSTGUSE_AGENT_THRESHOLD).....0
SYSTEM LEVEL THRESHOLD (WFSTGUSE_SYSTEM_THRESHOLD).....90

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

ICF CATALOG QUALIFIERS
-----
DBE1

SIZES PANEL 1 (DSNTIPD)
-----
USER LOB VALUE STORAGE IN KB (LOBVALA).....10,240
SYSTEM LOB VALUE STORAGE IN MB (LOBVALS).....4,096
MAXIMUM NUMBER OF LE TOKENS (LEMAX).....20
USER XML VALUE STG IN KB (XMLVALA).....204,800
SYSTEM XML VAL STG IN MB (XMLVALS).....10,240

PROTECTION PANEL (DSNTIPP1)
-----
SECURITY ADMINISTRATOR 1 AUTHORIZATION ID (SECADM1).....SECADM
SECURITY ADMINISTRATOR 1 TYPE (SECADM1_TYPE).....AUTHID
SECURITY ADMINISTRATOR 2 AUTHORIZATION ID (SECADM2).....SECADM
SECURITY ADMINISTRATOR 2 TYPE (SECADM2_TYPE).....AUTHID
SEPARATE SECURITY DUTIES (SEPARATE_SECURITY).....NO
INCLUDE DEPENDENT PRIVILEGES (REVOKE_DEP_PRIVILEGES).....SQLSTMT

DEFAULT STARTUP MODULES (DSNTIP03)
-----
PARAMETER MODULE.....DSNZPARM
ACCESS CONTROL (ACCESS_CNTL_MODULE).....DSNX0XAC
IDENTIFY/AUTH (IDAUTH_MODULE).....DSNX0ATH
SIGNON (SIGNON_MODULE).....DSNX0SGN

```

LOCATION: PMODBE1
 GROUP: DBE1
 MEMBER: SE11
 SUBSYSTEM: SE11
 DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
 SYSTEM PARAMETERS REPORT

PAGE: 1-6

ACTUAL FROM: 07/15/13 03:46:00.00

BUFFER POOL PARAMETERS

TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	5000
BUFFER POOL ID	BP0	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0
TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	10000
BUFFER POOL ID	BP1	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0
TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	20000
BUFFER POOL ID	BP2	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0

LOCATION: PMODBE1
 GROUP: DBE1
 MEMBER: SE11
 SUBSYSTEM: SE11
 DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
 SYSTEM PARAMETERS REPORT

PAGE: 1-7

ACTUAL FROM: 07/15/13 03:46:00.00

BUFFER POOL PARAMETERS

TIMESTAMP	BUFFER POOL ID	VPPOOL SIZE (PAGES)	VPPOOL SEQ THRESH	HORIZ DEFER WRITE THRESH	VERT DEFER WRITE THRESH (%)	VERT DEFER WRITE THRESH (BUF)	VPPOOL PARALLEL SEQ THRESH	ASSISTING PARALLEL SEQ THRESH	PGFIX ATTRIBUTE	PAGE STEAL METHOD	AUTOSIZE	FRAMESIZE	VPPOOL SIZE MIN	VPPOOL SIZE MAX
07/15/13 03:46:00.00	BP3	5000	80	30	5	0	50	0	NO	LRU	NO	4K	0	0
07/15/13 03:46:00.00	BP7	10000	80	30	5	0	50	0	NO	LRU	NO	4K	0	0
07/15/13 03:46:00.00	BP9	10	80	30	5	0	50	0	NO	LRU	NO	4K	0	0

LOCATION: PMODBE1
 GROUP: DBE1
 MEMBER: SE11
 SUBSYSTEM: SE11
 DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
 SYSTEM PARAMETERS REPORT

PAGE: 1-8

ACTUAL FROM: 07/15/13 03:46:00.00

BUFFER POOL PARAMETERS

TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	10
BUFFER POOL ID	BP10	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0
TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	35
BUFFER POOL ID	BP11	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0
TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	50
BUFFER POOL ID	BP12	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0

LOCATION: PMODBE1
 GROUP: DBE1
 MEMBER: SE11
 SUBSYSTEM: SE11
 DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
 SYSTEM PARAMETERS REPORT

PAGE: 1-9

ACTUAL FROM: 07/15/13 03:46:00.00

BUFFER POOL PARAMETERS

TIMESTAMP	BUFFER POOL ID	VPPOOL SIZE (PAGES)	VPPOOL SEQ THRESH	HORIZ DEFER WRITE THRESH	VERT DEFER WRITE THRESH (%)	VERT DEFER WRITE THRESH (BUF)	VPPOOL PARALLEL SEQ THRESH	ASSISTING PARALLEL SEQ THRESH	PGFIX ATTRIBUTE	PAGE STEAL METHOD	AUTOSIZE	FRAMESIZE	VPPOOL SIZE MIN	VPPOOL SIZE MAX
07/15/13 03:46:00.00	BP13	50	80	30	5	0	50	0	NO	LRU	NO	4K	0	0
07/15/13 03:46:00.00	BP14	50	80	30	5	0	50	0	NO	LRU	NO	4K	0	0
07/15/13 03:46:00.00	BP15	75	80	30	5	0	50	0	NO	LRU	NO	4K	0	0

LOCATION: PMODBE1
 GROUP: DBE1
 MEMBER: SE11
 SUBSYSTEM: SE11
 DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
 SYSTEM PARAMETERS REPORT

PAGE: 1-10

ACTUAL FROM: 07/15/13 03:46:00.00

BUFFER POOL PARAMETERS

TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	1000
BUFFER POOL ID	BP32K	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0
TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	5000
BUFFER POOL ID	BP32K1	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0
TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	10000
BUFFER POOL ID	BP8K0	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0

LOCATION: PMODBE1
 GROUP: DBE1
 MEMBER: SE11
 SUBSYSTEM: SE11
 DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
 SYSTEM PARAMETERS REPORT

PAGE: 1-11

ACTUAL FROM: 07/15/13 03:46:00.00

BUFFER POOL PARAMETERS

BUFFER POOL ID		VPPOOL SIZE (PAGES)	
TIMESTAMP	07/15/13 03:46:00.00	VPPOOL SEQ THRESH	5000
BUFFER POOL ID	BP8K1	HORIZ DEFER WRITE THRESH	80
		VERT DEFER WRITE THRESH (%)	30
		VERT DEFER WRITE THRESH (BUF)	5
		VPPOOL PARALLEL SEQ THRESH	0
		ASSISTING PARALLEL SEQ THRESH	50
		PGFIX ATTRIBUTE	0
		PAGE STEAL METHOD	NO
		AUTOSIZE	LRU
		FRAMESIZE	NO
		VPPOOL SIZE MIN	4K
		VPPOOL SIZE MAX	0
TIMESTAMP	07/15/13 03:46:00.00	VPPOOL SIZE (PAGES)	2000
BUFFER POOL ID	BP8K4	VPPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPPOOL SIZE MIN	0
		VPPOOL SIZE MAX	0
TIMESTAMP	07/15/13 03:46:00.00	VPPOOL SIZE (PAGES)	2000
BUFFER POOL ID	BP16K0	VPPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPPOOL SIZE MIN	0
		VPPOOL SIZE MAX	0

LOCATION: PMODBE1
 GROUP: DBE1
 MEMBER: SE11
 SUBSYSTEM: SE11
 DB2 VERSION: V11

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
 SYSTEM PARAMETERS REPORT

PAGE: 1-12

ACTUAL FROM: 07/15/13 03:46:00.00

BUFFER POOL PARAMETERS

TIMESTAMP	07/15/13 03:46:00.00	VPOOL SIZE (PAGES)	2000
BUFFER POOL ID	BP16K1	VPOOL SEQ THRESH	80
		HORIZ DEFER WRITE THRESH	30
		VERT DEFER WRITE THRESH (%)	5
		VERT DEFER WRITE THRESH (BUF)	0
		VPOOL PARALLEL SEQ THRESH	50
		ASSISTING PARALLEL SEQ THRESH	0
		PGFIX ATTRIBUTE	NO
		PAGE STEAL METHOD	LRU
		AUTOSIZE	NO
		FRAMESIZE	4K
		VPOOL SIZE MIN	0
		VPOOL SIZE MAX	0

LOCATION: PMODBE1
 GROUP: DBE1

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
 SYSTEM PARAMETERS REPORT

PAGE: 2-1

ACTUAL FROM: 07/15/13 03:46:00.00

DB2 VERSION: V11

GROUP BUFFER POOL PARAMETERS

TIMESTAMP	07/15/13 03:46:00.01	CURRENT DIRECTORY TO DATA RATIO	5
MEMBER	SE11	CLASS CASTOUT THRESHOLD (%)	5
GBP ID	GBP0	CLASS CASTOUT THRESHOLD (PAGES)	0
ALLOCATED GBP SIZE (4K)	768	GBP CASTOUT THRESHOLD (%)	30
ACTUAL DIRECTORY	1414	GBP CHECKPOINT INTERVAL (MIN)	4
ACTUAL DATA ENTRY	282	GBP CACHE SETTING	YES
PENDING DIRECTORY TO DATA RATIO	5	AUTO REC	YES
MODE	SIMPLEX		
TIMESTAMP	07/15/13 03:46:00.01	CURRENT DIRECTORY TO DATA RATIO	5
MEMBER	SE11	CLASS CASTOUT THRESHOLD (%)	5
GBP ID	GBP32K	CLASS CASTOUT THRESHOLD (PAGES)	0
ALLOCATED GBP SIZE (4K)	1536	GBP CASTOUT THRESHOLD (%)	30
ACTUAL DIRECTORY	231	GBP CHECKPOINT INTERVAL (MIN)	4
ACTUAL DATA ENTRY	44	GBP CACHE SETTING	YES
PENDING DIRECTORY TO DATA RATIO	5	AUTO REC	YES
MODE	SIMPLEX		
TIMESTAMP	07/15/13 03:46:00.01	CURRENT DIRECTORY TO DATA RATIO	5
MEMBER	SE11	CLASS CASTOUT THRESHOLD (%)	5
GBP ID	GBP8K0	CLASS CASTOUT THRESHOLD (PAGES)	0
ALLOCATED GBP SIZE (4K)	1280	GBP CASTOUT THRESHOLD (%)	30
ACTUAL DIRECTORY	1657	GBP CHECKPOINT INTERVAL (MIN)	4
ACTUAL DATA ENTRY	330	GBP CACHE SETTING	YES
PENDING DIRECTORY TO DATA RATIO	5	AUTO REC	YES
MODE	SIMPLEX		
TIMESTAMP	07/15/13 03:46:00.01	CURRENT DIRECTORY TO DATA RATIO	5
MEMBER	SE11	CLASS CASTOUT THRESHOLD (%)	5
GBP ID	GBP16K0	CLASS CASTOUT THRESHOLD (PAGES)	0
ALLOCATED GBP SIZE (4K)	1280	GBP CASTOUT THRESHOLD (%)	30
ACTUAL DIRECTORY	646	GBP CHECKPOINT INTERVAL (MIN)	4
ACTUAL DATA ENTRY	129	GBP CACHE SETTING	YES
PENDING DIRECTORY TO DATA RATIO	5	AUTO REC	YES
MODE	SIMPLEX		

Chapter 16. General Accounting information

This topic describes the Accounting report set and concepts in detail.

From the data shown in Accounting reports you can assign DB2 resource costs to individual authorization IDs and tune programs. OMEGAMON XE for DB2 PE does not, however, provide for tasks such as charge-back or billing.

Typical uses of Accounting reports are to:

- Summarize the local and distributed DB2 activity associated with the execution of specified DB2 plans.
- Summarize the composite DB2 activity associated with the execution of threads that exploit query CP parallelism or Sysplex query parallelism.
- Summarize the DB2 activity associated with the execution of the specified DB2 packages and DBRMs.
- Detect potential problem areas within DB2 applications.
- Track DB2 resource usage on a regular basis by OMEGAMON XE for DB2 PE identifiers such as location, authorization ID, or plan name. You can use this information to study trends or deviations from trends.
- Identify DB2 threads that fail to meet user-specified criteria.

In data sharing groups, you can produce Accounting reports at member level or at group level.

Accounting reports provide summary data about DB2 resource usage for:

- A given thread
- In thread reuse, the interval of time between two signons
- All tasks within a thread that exploit query CP or Sysplex query parallelism

Exception processing is supported in the Accounting report set. You define exception thresholds for certain Accounting fields. When you request exception processing, the values in these fields are checked against the thresholds. Only those records that contain field values outside the defined thresholds are reported.

TOP processing identifies the main consumers of DB2 resources. If the TOP subcommand option is used, an index-like report is generated that shows the main consumers at the end of the report or trace. If you only want to see the main resource consumers, specify the TOP subcommand option with the ONLY keyword.

“Functions and utilities of the Accounting report set” on page 138

The Accounting report set consists of the REPORT, TRACE, FILE, REDUCE, SAVE, RESTORE functions, and the Save-File and Spreadsheet Input-Data Generator utilities.

“Thread types” on page 139

The DB2 thread or, for a thread in reuse, the part of it that is between two consecutive signons or resignon, is the basic unit of reporting for the Accounting report set.

“Distributed activity” on page 142

The Accounting trace provides insights to activities among distributed DB2 subsystems as well as non-DB2 requestors.

“Query parallelism considerations” on page 143

The Accounting report set supports parallel tasks from CP parallelism and Sysplex query parallelism.

“Timing information” on page 145

The Accounting report set provides a multitude of timing information for nondistributed and distributed transactions.

“Input for Accounting reports” on page 148

The Accounting report set can process data originating at different DB2 locations in the same OMEGAMON XE for DB2 PE run.

Functions and utilities of the Accounting report set

The Accounting report set consists of the REPORT, TRACE, FILE, REDUCE, SAVE, RESTORE functions, and the Save-File and Spreadsheet Input-Data Generator utilities.

REPORT

Shows Accounting data summarized by OMEGAMON XE for DB2 PE identifiers. For example, you can produce a report that shows the Accounting information for threads, plans, or users. The report shows totals and subtotals of fields.

For a DB2 system in a data sharing group, reports can be at member or group level.

TRACE

Shows Accounting data for a particular thread. Unlike reports, traces show the Accounting data without aggregation. This means that the records are listed individually, in the order of occurrence. There is an exception if a thread exploits parallelism. In this instance, all parallel activity is aggregated and presented as a single trace entry.

Traces can be very long. Use traces only to resolve a specific problem.

FILE Formats DB2 Accounting records and stores the records in sequential data sets that can be loaded into DB2 tables. You can analyze the data in DB2 tables and produce tailored reports by using a reporting facility such as Query Management Facility (QMF). You can also use FILE to produce data sets that contain only exception records. (See “Creating data for the Performance Database and the Performance Warehouse” on page 273.)

You can also use the File data sets to generate CSV (comma-separated value) input-data. This CSV data can then be transferred to workstations and imported into spreadsheets to improve DB2 performance analysis using graphical representations or pivot tables. (See Chapter 25, “Generating input data for spreadsheets,” on page 275.)

REDUCE

Reduces the volume of data that is input to the REPORT and SAVE functions. REDUCE is invoked automatically if you use REPORT or SAVE. Specify REDUCE explicitly if you want to:

- Specify an interval to order data on Accounting reports
- Produce several reports to cover different time periods

After reducing data, the resulting data set is much smaller than the original input data set. However, reducing data uses a considerable amount of system resources. (See “Creating data for the Performance Database and the Performance Warehouse” on page 273.)

SAVE You can save data sets as follows:

Without CONVERT option

Produces VSAM data sets that contain reduced Accounting records. When the data is saved, you can:

- Combine it with new data to produce long-term reports.
- Use the Save-File utility to create sequential data sets that are suitable for use by the DB2 load utility.

You can use REDUCE and SAVE to keep historical Accounting data about DB2 performance. You can define the interval and the input filters for the data that is saved into the Save data set.

With CONVERT option

Produces physical sequential data sets that contain reduced and converted Accounting records. When the data is saved, it can be used by the DB2 load utility.

You can also use the Save-File data sets to generate CSV (comma-separated value) input-data. This CSV data can then be transferred to workstations and imported into spreadsheets to improve DB2 performance analysis using graphical representations or pivot tables. (See Chapter 25, “Generating input data for spreadsheets,” on page 275.)

RESTORE

Reloads a previously saved VSAM data set for additional use. This saved data can be restored and re-saved as often as required.

Save-File utility

With the Save-File utility you can:

- Migrate Accounting Save data sets from earlier releases.
- Convert Accounting Save data sets into sequential data sets that are suitable for use by the DB2 load utility.

(See “Creating data for the Performance Database and the Performance Warehouse” on page 273.)

Spreadsheet Input Data Generator utility

You can also use the File and Save-File data sets to generate CSV (comma-separated value) input-data. This CSV data can then be transferred to workstations and imported into spreadsheets to improve DB2 performance analysis using graphical representations or pivot tables. (See Chapter 25, “Generating input data for spreadsheets,” on page 275.)

Thread types

The DB2 thread or, for a thread in reuse, the part of it that is between two consecutive signons or resignon, is the basic unit of reporting for the Accounting report set.

Thread categories

OMEGAMON XE for DB2 PE uses the following categorization of DB2 threads:

Allied thread

An allied thread:

- Originates at the local DB2 subsystem and can access data at a remote DB2 subsystem.

- Does not involve distributed activity, that is, it is not initiated by a remote location and does not request data from another location.
- Can be filtered for Accounting by selecting the OMEGAMON XE for DB2 PE identifier THREADTYPE with a value of ALLIED. The Accounting report ordered by THREADTYPE shows ALLIED as an individual criterion.

The Accounting record that represents an allied thread consists of the following set of data, which is called *non-DDF (Distributed Data Facility) data*:

- Identification of the thread
- General timing
- SQL and RID list usage
- Query parallelism
- Buffer pool activity
- Group buffer pool activity
- Data sharing locking
- Stored procedures
- Data capture
- Locking activity
- Packages and DBRMs executed
- Resource limit facility data

Allied-distributed thread

An allied-distributed thread:

- Requests work from remote server locations.
- Is not initiated by a remote location.
- Can be filtered for Accounting by selecting the OMEGAMON XE for DB2 PE identifier THREADTYPE with a value of ALLIED_DIST. The Accounting report ordered by THREADTYPE can show ALLDDIST as an individual criterion.

The Accounting record that represents an allied-distributed thread consists of:

- Non-DDF data
- One block of *DDF data* for each participating server location, which includes information such as the number of messages, statements, rows, and bytes that have been sent and received.

Database access thread (DBAT)

A DBAT thread:

- Accesses data at the local subsystem on behalf of a remote subsystem.
- Can be filtered for Accounting by selecting the OMEGAMON XE for DB2 PE identifier THREADTYPE with a value of DBAT.

The Accounting record that represents a DBAT consists of:

- Non-DDF data
- DDF data for the requester location
- Also includes DBAT-distributed threads that are initiated by a requester location and executed by the server location that in turn requests data from another server location.

The Accounting report ordered by THREADTYPE can show one or several of the following criterion:

DBAT Indicates accumulated data of threads that are initiated, created, and performing work on behalf of a remote (requester) location.

DBATDP

Indicates accumulated data of DBAT duplicate threads.

DBATDIST

Indicates accumulated data of DBAT distributed threads that are initiated by a requester location and executed by the server location that in turn requests data from another server location.

DBATDICP

Indicates accumulated data of DBAT distributed and copy threads.

DBATDIDP

Indicates accumulated data of DBAT distributed and duplicate threads.

For example, when location A uses DRDA[®] to request data at location B and, in the same unit of work, accesses data at location C (using DB2 private protocol), the thread created at location A is an allied-distributed thread, the thread created at location B is a DBAT-distributed thread. The thread created at location C is a DBAT.

The Accounting record that represents a DBAT-distributed thread consists of:

- Non-DDF data
- DDF data for the requester location
- One block of DDF data for each participating server location

Thread types reported by ORDER

The following terms can help you to understand the concepts of the different thread types and merged processing:

Nondistributed transaction

A nondistributed transaction for Accounting:

- Is initiated by DB2 and performed at one location without interaction with other locations. For example, if an allied thread is not reused, it represents a nondistributed transaction. If it is reused, a nondistributed transaction is a DB2 activity between two signons.
- Can be filtered by specifying the INCLUDE or EXCLUDE subcommand options with the OMEGAMON XE for DB2 PE identifier THREADTYPE using a value of ALLIED or DBAT. As a result the Accounting report ordered by THREADTYPE can show a thread type of ALLIED, DBAT, or DBATDP.

Note: The OMEGAMON XE for DB2 PE identifier DBAT also covers the distributed transactions.

Distributed transaction

A distributed transaction for Accounting:

- Is initiated by DB2 at one (requester) location and performed at one or more remote (server) locations.
- Consists of a local activity that is represented by an allied-distributed thread, and in case of a loopback from a DBAT, remote activity that is represented by one or more DBATs. Therefore, a distributed transaction requires Accounting records for the allied-distributed thread and all corresponding DBATs.

- Can be filtered by specifying the INCLUDE or EXCLUDE subcommand options with the OMEGAMON XE for DB2 PE identifier THREADTYPE using a value of ALLIED-DIST or DBAT.

Note: The OMEGAMON XE for DB2 PE identifier DBAT also covers the nondistributed transactions.

The Accounting report ordered by THREADTYPE can show one or several of the following thread types that are distributed transactions:

ALLDDIST

Indicates accumulated data of threads initiated by DB2 and that request data from one or more server locations.

DBATDIST

Indicates accumulated data of DBAT distributed threads that are initiated by a requester location and executed by the server location that in turn requests data from another server location.

DBATDICP

Indicates accumulated data of DBAT distributed and copy threads.

DBATDIDP

Indicates accumulated data of DBAT distributed and duplicate threads.

Activity location

Reports and traces are location-oriented. They show activity that is performed at one or more locations. For a given location, the following information is shown:

- The nondistributed transactions, in other words, the allied threads at that location
- The local activity of distributed transactions that originate at that location, in other words, the allied-distributed threads from that location without the corresponding DBATs at other locations
- The remote activity at that location as part of distributed transactions requested from other locations, in other words, the DBATs at that location

Multi-site or single-site reports

Reports and traces can be single-site or multi-site:

- Single-site reports and traces present Accounting data for one location. You can obtain a single-site report or trace by processing input data that only contains records from a single location or by specifying a single location with the INCLUDE or EXCLUDE subcommand options.
- Multi-site reports and traces present Accounting data for more than one location. The data is arranged in alphabetical order by location name.

Distributed activity

The Accounting trace provides insights to activities among distributed DB2 subsystems as well as non-DB2 requestors.

OMEGAMON XE for DB2 PE supports communication between:

- DB2 for z/OS subsystems
- DB2 for z/OS subsystem and non-DB2 requestors, such as SQL/DS, or ORACLE

For detailed information about the communication between these systems, produce an Accounting trace.

For communication between two DB2 subsystems, the Accounting trace provides you with information about the requester locations (for DBATs) and the server locations (for requester threads).

For communication between a DB2 subsystem and a non-DB2 for z/OS requester, OMEGAMON XE for DB2 PE can only present performance data on DBATs. However, the Accounting trace helps you to identify the requester. It supplies the identifier and release level of the requester involved. For requesters from DB2 on other platforms also provides the client platform, application name, authorization ID, and a user-supplied part. If possible, OMEGAMON XE for DB2 PE uses the first eight bytes of the application name to identify the DBAT. If it is not possible to show the DBAT application name, the plan name is shown as DISTSERV. When DISTSERV is shown, you can use the MAINPACK identifier to distinguish DBATs.

Accounting trace records are affected by the DB2 subsystem parameter ACCUMACC, which controls whether and when DB2 Accounting data is accumulated by the user for DDF and RRSF threads. A parameter value of 2 or greater causes Accounting records to roll up into a single record every *n* occurrences of the user on the thread. These values can be set by DDF threads by Server Connect and Set Client calls, and by RRSF threads by RRSF SIGN, AUTH SIGNON, and CONTEXT SIGNON functions. When roll-up occurs, the values of some fields shown in Accounting reports and traces lose their meanings because of the accumulation. Thus, these fields are marked as N/P or N/C for derived fields. For a list of affected fields, see “Fields affected by roll-up during distributed and parallel tasks” on page 287.

Query parallelism considerations

The Accounting report set supports parallel tasks from CP parallelism and Sysplex query parallelism.

If a query exploits query CP parallelism or Sysplex query parallelism, several tasks (called parallel tasks) perform the work. For each of these tasks an Accounting record is generated, which contains counters and timers pertinent to the work performed by the particular task. In addition, an Accounting record is created that contains the details about nonparallel work within the thread and data related to parallel work.

OMEGAMON XE for DB2 PE summarizes all Accounting records that are generated for such a query and presents the records as one logical Accounting record. Table 16 describes which values are taken from both the originating and parallel records and which are taken from the originating record only.

Table 16. Data related to query CP and Sysplex query parallelism

Accounting data	Derivation
Identifiers (such as PRMAUTH or PLANNAME)	Originating record
Class 1 elapsed times	Originating record
Class 1 TCB times	Originating and parallel records
Class 2 elapsed times	Originating record
Class 2 TCB times	Originating and parallel records

Table 16. Data related to query CP and Sysplex query parallelism (continued)

Accounting data	Derivation
Class 7 elapsed times	Originating record
Class 7 TCB times	Originating and parallel records
Class 2 and 7 DB2 entry/exit events	Originating record
Class 3 and 8 times and events	Originating and parallel records
Class 5 times	Originating record
SQL counters	Originating record
RID list counters	Originating and parallel records
Query parallelism counters	Originating record
Locking (including data sharing) counters	Originating and parallel records
RLF data	Originating record
Buffer pool counters	Originating and parallel records
Group buffer pool counters	Originating and parallel records
DDF counters	Originating record
Data capture counters	Originating record
SU counters	Originating and parallel records

The elapsed time is taken from the originating record, while CPU and suspension times are calculated from all parallel and originating records. Consequently, both CPU time and suspension times can be larger than the elapsed time. Therefore, you can only get the full picture of response time distribution if the times for each participating task are known. If you suspect that the CPU times or suspension times for a thread where query CP or Sysplex query parallelism is used are, for other reasons, larger than the times being added for several tasks, produce a long Record trace for IFCID 3. This way all parallel and originating Accounting records are reported separately.

In Sysplex query parallelism, the CPU times of the parallel records are *normalized* so that you can add up the times across multiple DB2s running on different machines. Normalized means that the CPU times are converted to a common unit, called *service unit (SU)*, using a conversion factor. The conversion factor depends on the machine being used.

By default, Accounting trace records from parallel query tasks are rolled up into the originating task's Accounting trace. This is controlled by the DB2 subsystem parameter PTASKROL (parallel tasks roll-up), which has a default value of YES. A roll-up record is written when the parent task (agent) deallocates on an originating DB2, or when an accumulating child task is deallocated on an assisting DB2. The rolled up data is an accumulation of all counters for that field for each child task that completed and deallocated. When roll-up occurs, the values of some fields in Accounting reports and traces lose their meanings because of the accumulation. Thus, these fields are marked as N/P, or N/C for derived fields. For a list of affected fields, see "Fields affected by roll-up during distributed and parallel tasks" on page 287.

Timing information

The Accounting report set provides a multitude of timing information for nondistributed and distributed transactions.

OMEGAMON XE for DB2 PE reports the following timing information:

- Application times (class 1)
- DB2 times (class 2)
- Suspension times (class 3) and counts
- IFI times (class 5)
- DB2 times on a per package/DBRM basis (class 7)
- Suspensions times on a per package/DBRM basis (class 8) and counts

If an Accounting record represents an allied-distributed thread, additional time fields for DDF data are reported. These DDF times are obtained from Accounting class 1.

The following figure shows an example of the DB2 Accounting times for a nondistributed transaction that does not exploit CP parallelism and how the Accounting times relate to each other and to the DB2 SQL call activity. The example depicts an application for IMS or CICS. Classes 5, 7, 8, and 9 are not shown. Class 5 is a subset of class 2 and is present only if IFI processing takes place. Classes 7 and 8 are equivalent to classes 2 and 3, but on a package or DBRM level.

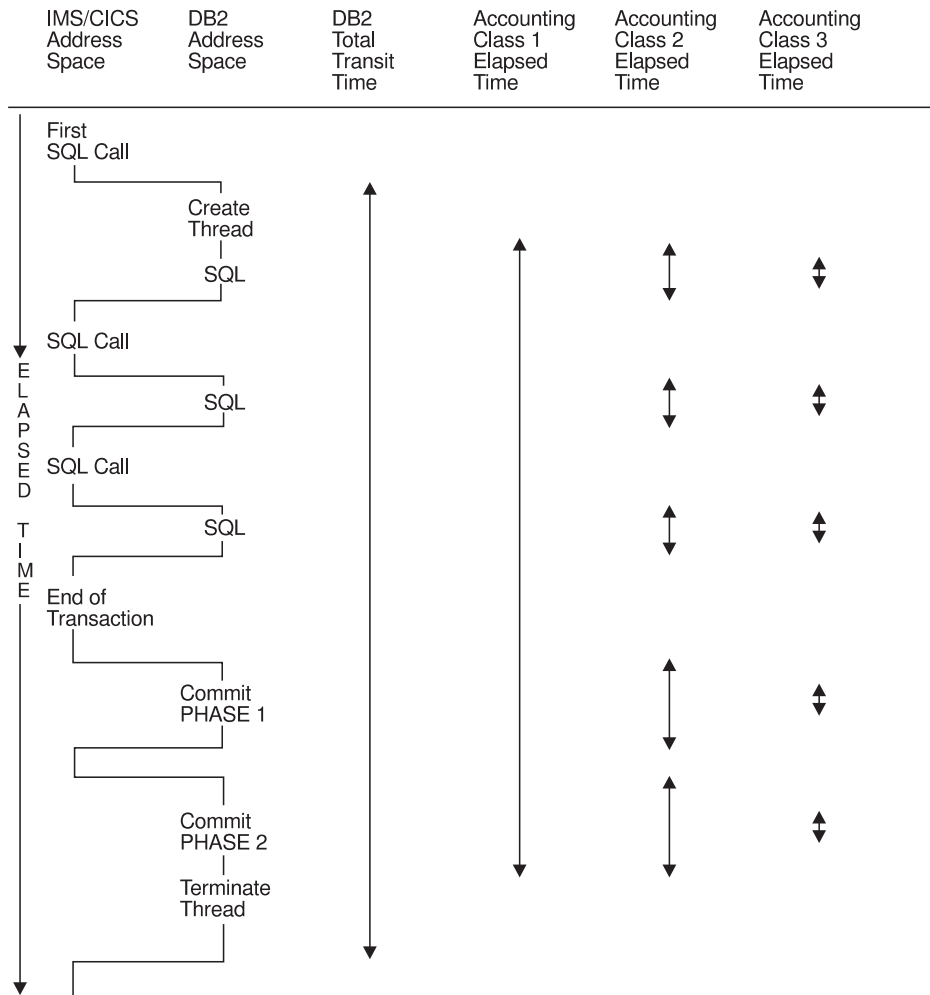


Figure 39. DB2 Accounting times for a nondistributed transaction

The following figure shows various elapsed times for a distributed transaction originating from TSO allied space. CP parallelism is not exploited. The times for both the requester and the server are shown. This figure is a simplified presentation of the processes at the participating locations. It does not, for example, show block fetch statements and it is only applicable to a single row retrieval. DDF-at-Server time is not provided if the transaction uses DRDA.

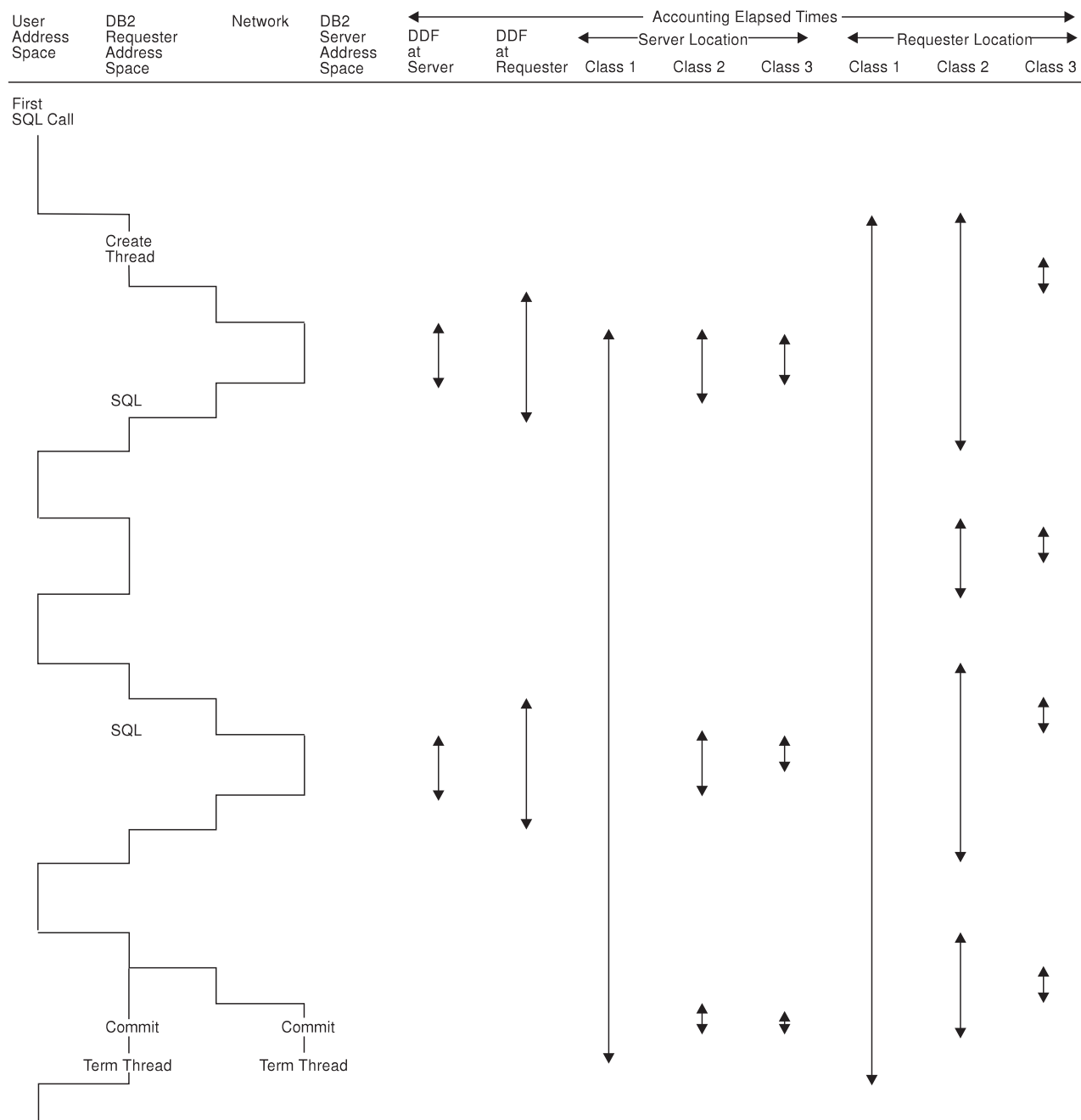


Figure 40. Accounting elapsed times for a distributed transaction

The various elapsed times in the figure are defined as follows:

- DDF at Server

This is the ELAPSED SER field that is reported in the Requester Fields for Server Location section of the Accounting reports and traces. It represents the elapsed time spent at the server between the time the SQL statement is received and the time the answer is sent to VTAM®. This time is not applicable to DRDA and N/A is printed.

- DDF at Requester

This is the ELAPSED REQ field in the Server Fields for Requester Location section of the Accounting reports and traces. It represents the elapsed time spent at the requester between the time the SQL statement is sent and the time the answer from the server is received.

- Class 1 at Server Location

This is the ELAPSED TIME field under the APPLICATION TIMES (CLASS 1) column of the Accounting reports and traces for the server location. It represents the class 1 elapsed time from creation to termination of the DBAT.

- Class 2 at Server Location

This is the ELAPSED TIME field under the DB2 TIMES (CLASS 2) column of the Accounting reports and traces for the server location. It represents the elapsed time to process the SQL requests and the commits at the server.

- Class 3 at Server Location

This is the ELAPSED TIME for the total of the class 3 suspensions in the Accounting reports and traces for the server location. It represents the time the DBAT was suspended while waiting for various system events such as locking, I/O, and other requests.

- Class 3 at Requester Location

This is the ELAPSED TIME column for class 3 suspensions on the Accounting reports and traces for the requester location. It represents the time the allied-distributed thread was suspended while waiting for various system events such as locking, I/O, and other requests.

- Class 2 at Requester Location

This is the ELAPSED TIME field under the DB2 TIMES (CLASS 2) column of the Accounting reports and traces for the requester location. It represents the elapsed time between the application passing the SQL statement to the local (requester) DB2 system and back again. This is the time spent in DB2.

- Class 1 at Requester Location

This is the ELAPSED TIME field under the APPLICATION TIMES (CLASS 1) column of the Accounting reports and traces for the requester location. It represents the elapsed time from creation to termination of the allied-distributed thread.

Input for Accounting reports

The Accounting report set can process data originating at different DB2 locations in the same OMEGAMON XE for DB2 PE run.

Several input data sets (in any SMF, GTF, DPMOUT, and Online Monitor trace output data set format) can be logically concatenated in the DD statements for INPUTDD. The data is sorted in the primary sequence of location and reported according to the parameters specified in the Accounting command.

“DB2 traces used in Accounting”

The Accounting report set requires the following DB2 trace classes and DB2 trace types to be active.

“Accounting record generation” on page 149

About how and when Accounting data is collected and recorded.

“Missing data sections” on page 150

Conditions that might cause missing data for Accounting reports.

DB2 traces used in Accounting

The Accounting report set requires the following DB2 trace classes and DB2 trace types to be active.

Accounting uses the DB2 IFCIDs 3 and 239 (DB2 trace type Accounting, DB2 trace class 1) as input for the reports and traces. Timing data is taken from all classes:

- Application times (class 1)
- DB2 times (class 2)
- Suspension times (class 3) and counts
- IFI times (class 5)
- DB2 times on a per package/DBRM basis (class 7)
- Suspensions times on a per package/DBRM basis (class 8) and counts

If an Accounting record represents an allied-distributed thread, additional time fields for DDF data are reported. These DDF times are obtained from Accounting class 1.

Accounting trace classes are started either at DB2 start time, by using the installation panel DSNTIPN, or by entering the DB2 **START TRACE** command at the terminal. For example:

```
-START TRACE(ACCTG) CLASS(1,2,3,5,7,8) DEST(SMF)
```

The Accounting classes 1 and 3 should always be active. The overhead is not significant, whereas the information they provide is crucial for all aspects of performance monitoring. It is also useful to always have class 2 active. If you do not always have class 3 active, activate class 3 when you activate class 2.

Classes 7 and 8 provide valuable information on a per package or per DBRM basis. They are equivalent to classes 2 and 3. Package or DBRM Accounting is probably most beneficial for a DB2 server of a non-DB2 requester or when an application plan can execute many packages or DBRMs, but actually executes 10 or less for a given Accounting record. If class 2 is active, also activate class 7 if package Accounting is needed. If class 3 is active, also activate class 8 if package Accounting is needed. More than 10 packages or DBRMs introduce additional overhead in data collection reducing the benefit versus cost. However, this cost is still significantly less than a DB2 performance class trace.

Accounting record generation

About how and when Accounting data is collected and recorded.

The collection of Accounting data begins when a thread connects to DB2. A completed Accounting record is written when:

- The thread terminates
- The authorization identifier changes because the thread is reused
- A DBAT becomes inactive
- A parallel task completes when query CP or Sysplex query parallelism is exploited

Note:

- NEW USER is reported when DB2 Accounting records are reported in the IMS thread and CICS thread reuse situations where a new AUTHID is used.
- If TOKEN=YES on the TYPE=ENTRY statement in the RCT table is specified, the CICS attachment facility requests an Accounting record to be produced during thread reuse even if the user authorization ID does not change. A CICS LU6.2 token is also passed to DB2 allowing correlating CICS and DB2 trace records.
- For thread reuse, all values are accumulated since the last Accounting record.

- When a CICS application program causes more than one SYNC point commit or rollback, DB2 can produce several out-of-sequence Accounting records for the application if CICS attach thread swapping occurs.
- If a thread is reused with the same user authorization ID and TOKENS has not been specified, the DB2 Accounting record represents several CICS transactions.

Missing data sections

Conditions that might cause missing data for Accounting reports.

The following DB2 threads might result in missing data sections because required data could not be gathered:

- If attaching to QMF that generates a unique DB2 thread.
Under this thread no SQL, buffer, or locking information is gathered.
- If running a dynamic SQL that generates a unique DB2 thread.
Under this thread, SQL, buffer, and locking activity can be performed. A data section is produced if activity in the data section occurs. Resource limit activity is only reported in the Accounting record for dynamic SQL activity.
- If ending a QMF session that creates a unique DB2 thread.
Under this thread no SQL, buffer, or locking information is gathered.
- If running a batch job that creates a unique DB2 thread.
Under this thread SQL, buffer, and locking activity can be performed. No resource limit activity is tracked for this thread.
- If location A uses DRDA protocol to access data at location B and does not do any local work, no SQL information is gathered for the corresponding allied-distributed thread at location A.

Chapter 17. Creating effective Accounting reports

For a report to be effective it must deliver the information you need efficiently in terms of resources required to produce it, the volume of data produced, and the time it takes for you to interpret the report.

“Processing considerations”

Because Accounting is one of the most frequently used report sets, consider the effects on batch resources. To reduce processing time when producing Accounting reports and traces, only ask for the information that you really need.

“Exception processing” on page 158

Use exception processing to identify Accounting report, trace, and File data set entries with fields that contain values outside thresholds specified in the Exception Threshold data set. When exception processing is active, the data to be reported is checked against these thresholds. You can obtain exception reports, traces, and File data sets by using the EXCEPTION subcommand option in ACCOUNTING subcommands.

“ORDER processing” on page 160

You can change the way reports are summarized by using the ORDER subcommand option of the REPORT subcommand. The ORDER subcommand option specifies which OMEGAMON XE for DB2 PE identifiers are used to aggregate records. It also identifies the presentation sequence of the report entries, unless the TOP(ONLY) subcommand option is specified for Accounting.

“TOP processing” on page 171

TOP processing is useful to identify the report entries that might indicate a problem application.

“Reducing data” on page 173

Use the REDUCE subcommand to consolidate records with certain common characteristics into a single record.

“Member-scope and group-scope reporting” on page 185

DB2 enables a query to be processed by several members of a data sharing group. Each member can split the work into parallel tasks. Accounting aggregates the parallel activity done in each member and includes it in the originating query activity.

Processing considerations

Because Accounting is one of the most frequently used report sets, consider the effects on batch resources. To reduce processing time when producing Accounting reports and traces, only ask for the information that you really need.

Before producing a report or trace, consider how much detail you need.

Use the short (default) layouts of reports and traces whenever possible. In most instances these provide enough detail for monitoring and problem determination.

A detailed report that uses all input data that was gathered for a long period uses a lot of system resources. The result is pages of information that you are probably not interested in.

To avoid unnecessary processing overhead and to save time:

- Consider carefully how detailed a report you need.
- Filter the input data – preferably by using the GLOBAL command.
- Disable OMEGAMON XE for DB2 PE internal sort if appropriate.
- Define groups for identifiers that you want reported as a single entry.
- Specify a REDUCE INTERVAL only to report by intervals or to produce several reports with different time spans.
- Define exception thresholds only for fields that you are interested in.
- Specify DPMOUT or keep a Save data set only if you are sure that you want to report the data again.

“Choosing the right level of detail”

Before you produce a report or trace, consider how much detail you need. Use the LAYOUT subcommand option of the Accounting reports and traces to control the amount of data to be produced.

“Filtering data” on page 153

You can limit the amount of data to be processed by filtering the input data. You can specify filters in the GLOBAL command and in the REDUCE, REPORT, TRACE, and FILE subcommands.

“Suppressing internal sort” on page 154

When you request Accounting functions only, it is often possible to avoid the OMEGAMON XE for DB2 PE internal sort of the input data. Suppressing the internal sort with the PRESORTED option of the GLOBAL command reduces the size of the sort work files that must be allocated and the processing time.

“Grouping data” on page 154

Use the GROUP command to collect several OMEGAMON XE for DB2 PE identifier values under one name. When you request a report and specify this name by using the INCLUDE or EXCLUDE subcommand option, the events for all individual items are consolidated into one.

“Specifying intervals for long-term reporting” on page 157

If you want to perform a trend analysis, you probably want to report data by interval. To do this, first reduce the input data for your Accounting reports using an interval other than the default zero. Then, order the report by intervals.

“Specifying exception thresholds for specific fields” on page 157

You can set exception thresholds for virtually all Accounting fields. However, carefully consider the fields for which to specify exception thresholds. The more fields you specify, the greater the effects on processing.

“Using a DPMOUT data set” on page 157

Keep the content of the DPMOUT data set for further reports.

“Saving reduced data” on page 157

This section shows alternative ways to save reduced data. Use the subcommands REDUCE and SAVE (without CONVERT), for example, to store historical Accounting data in smaller data sets.

Choosing the right level of detail

Before you produce a report or trace, consider how much detail you need. Use the LAYOUT subcommand option of the Accounting reports and traces to control the amount of data to be produced.

In most situations the default layouts, which are short versions of reports and traces, provide enough detail for monitoring and problem determination. Do not use the most comprehensive layouts that show all possible fields unless you need this information.

For Accounting the LAYOUT subcommand option ACCEL provides detailed thread-related Accelerator activity data.

If the use of commands, subcommands, and subcommand options does not provide sufficient control over the contents of reports and traces, you can use the User-Tailored Reporting (UTR) feature to create and tailor your own report and trace layouts. With UTR you can add, remove, and change individual fields and entire report blocks to control the volume, contents, and layouts of your reports and traces. For more information, see “Tailoring report layouts” on page 236.

Filtering data

You can limit the amount of data to be processed by filtering the input data. You can specify filters in the GLOBAL command and in the REDUCE, REPORT, TRACE, and FILE subcommands.

Specify the filters in GLOBAL whenever you can, because only the data that passes through the GLOBAL filters is processed further. The less data OMEGAMON XE for DB2 PE needs to process, the better the performance.

However, ensure that you do not exclude records that are needed in OMEGAMON XE for DB2 PE processing.

“FROM and TO subcommand options”

The simplest filter is the start and end date and time of the data to be reported. Specify the start and end dates and times by using the FROM and TO subcommand options.

“INCLUDE and EXCLUDE subcommand options” on page 154

Another way to filter data is to include data only for particular OMEGAMON XE for DB2 PE identifier values, for example, user IDs or plans. You can do this by using the INCLUDE and EXCLUDE subcommand options.

FROM and TO subcommand options

The simplest filter is the start and end date and time of the data to be reported. Specify the start and end dates and times by using the FROM and TO subcommand options.

For example, to monitor the performance of your system only during peak hours and to produce the default version of a report, specify:

```
⋮
GLOBAL
  FROM (09/25/13,08:30)
  TO   (09/25/13,17:00)
ACCOUNTING
⋮
```

The report shows information from 8:30 a.m. to 5:00 p.m. for the specified day.

Presuming that the input data set contains data for more than one day, for example a week, you can generate a report that shows the performance of your system during peak hours for the whole week by specifying:

```
⋮
GLOBAL
  FROM (,08:30)
  TO   (,17:00)
```

```
ACCOUNTING
:
```

The report shows information from 8:30 a.m. to 5:00 p.m. for every day of the week.

INCLUDE and EXCLUDE subcommand options

Another way to filter data is to include data only for particular OMEGAMON XE for DB2 PE identifier values, for example, user IDs or plans. You can do this by using the INCLUDE and EXCLUDE subcommand options.

For example, if you have a problem with applications coming from one location, USIBMSNEWY11, and you know the CICS transactions are not causing a problem, specify:

```
:
```

```
GLOBAL
  INCLUDE (LOCATION (USIBMSNEWY11))
  EXCLUDE (CONNTYPE (CICS))
:
```

Continuing the example, suppose the report indicated a problem with authorization identifier USERID01 using plan PVLDD4C3. You are only interested in data belonging to that user ID and plan. You can now specify:

```
:
```

```
GLOBAL
  INCLUDE (LOCATION (USIBMSNEWY11))
  INCLUDE (AUTHID (USERID01))
  INCLUDE (PLANNAME (PVLDD4C3))
:
```

Suppressing internal sort

When you request Accounting functions only, it is often possible to avoid the OMEGAMON XE for DB2 PE internal sort of the input data. Suppressing the internal sort with the PRESORTED option of the GLOBAL command reduces the size of the sort work files that must be allocated and the processing time.

The PRESORTED option of the GLOBAL command controls the internal sort. For example, to produce an Accounting report without sorting the input data, specify:

```
:
```

```
GLOBAL
  PRESORTED (ACCEPT)
```

```
ACCOUNTING
:
```

The resulting Accounting report shows Accounting data for all locations in the input data set, without performing an internal sort.

Grouping data

Use the GROUP command to collect several OMEGAMON XE for DB2 PE identifier values under one name. When you request a report and specify this name by using the INCLUDE or EXCLUDE subcommand option, the events for all individual items are consolidated into one.

For more information on the GROUP command in Accounting refer to IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS; IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS: Report Command Reference.

“Grouping data using sets of identifiers”

You can use the GROUP command to define a set of OMEGAMON XE for DB2 PE identifier values that can be used when you request certain reports. The information for the set is reported as a single entry in the reports.

“Grouping data using the connection type identifier”

You can use the GROUP command to reduce data according to the connection type of the thread.

“Grouping data using the thread type identifier” on page 156

You can use the GROUP command to define and pass individual thread types.

“Grouping data by ordering according to thread types” on page 156

You can use the GROUP command to order data according to thread types.

Grouping data using sets of identifiers

You can use the GROUP command to define a set of OMEGAMON XE for DB2 PE identifier values that can be used when you request certain reports. The information for the set is reported as a single entry in the reports.

In GROUP processing the data for all items of the set is consolidated into one record. This improves OMEGAMON XE for DB2 PE performance because less records must be processed.

Sets are also useful for reporting purposes such as to report data for an entire department instead of every individual person.

Assume that your sales department consists of three users (USER001, USER002, and USER003) and you want to specify sets to produce an Accounting report with performance data for this department. You can specify:

```
⋮
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.

Grouping data using the connection type identifier

You can use the GROUP command to reduce data according to the connection type of the thread.

First you must define the GROUP command for the identifier CONNTYPE, such as GROUP (CONNTYPE (ALLCTP(*))). Then use the REDUCE subcommand to get SAVE records with the variable group name. The conversion utility can handle the group name and passes it to the Save-File utility (including the predefined names of the connection types).

In the following example, data for the connection type group named ALLCTP is accumulated. Records created in the SAVE data set with DD name ACSAVDD1 contain name ALLCTP as part of their key. After converting this data, the loadable data contain ALLCTP as a connection type.

```

:
:
GROUP (CONNTYPE (ALLCTP(*)))
ACCOUNTING
  REDUCE
    INCLUDE (CONNTYPE (G(ALLCTP)))
  SAVE
    DDNAME(ACSAVDD1)
:
:

```

Grouping data using the thread type identifier

You can use the GROUP command to define and pass individual thread types.

Specify the thread type you want to include in your report. The THREADTYPE in a report can be ordered by each individual thread type that needs to be distinguished when processing the REDUCE subcommand.

A logical report is created for each thread type and for each unique member (identified by DB2 location, group, subsystem, member) or for each DB2 group.

Note: The detailed thread types are only shown in the output, which can be an Accounting report or a loadable PDB file. This ensures that individual and separated data (such as two logical reports) do not refer to the same set name (such as DBAT).

Grouping data by ordering according to thread types

You can use the GROUP command to order data according to thread types.

The following example first defines the group, such as GROUP (THREADTYPE(THRTGRP(*))). Then the REDUCE subcommand is specified to reduce the usage of the group to report (ordered by the thread type) and save:

```

:
:
GROUP (THREADTYPE(THRTGRP(*)))
ACCOUNTING
  REDUCE
    INCLUDE (THREADTYPE (G(THRTGRP)))
  REPORT
    ORDER(THREADTYPE)
  SAVE
    DDNAME(ACSAVDD1)
:
:

```

Sample with GROUP REDUCE for THREADTYPE

The following example shows the part of a sample Accounting report that results from the command GROUP REDUCE ordered by THREADTYPE.

LOCATION: DEBDZ99NM222DSNY	OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)	PAGE: 1-1
GROUP: N/P	ACCOUNTING REPORT - LONG	REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P		TO: NOT SPECIFIED
SUBSYSTEM: DSNY	ORDER: THRTYPE	INTERVAL FROM: 11/26/11 23:00:49.00
DB2 VERSION: V10	SCOPE: MEMBER	TO: 11/27/11 22:59:55.00
THRTYPE: THRTGRP		

ELAPSED TIME DISTRIBUTION	CLASS 2 TIME DISTRIBUTION
-----	-----
APPL ===== 100%	CPU ====> 6%
DB2	SECPU ====> 11%
SUSP	NOTACC ====> 12%
	SUSP ===== 71%

AVERAGE	APPL(CL.1)	DB2 (CL.2)	IFI (CL.5)	CLASS 3 SUSPENSIONS	AVERAGE TIME	AV.EVENT	HIGHLIGHTS
-----	-----	-----	-----	-----	-----	-----	-----
ELAPSED TIME	52.522821	0.022607	N/P	LOCK/LATCH(DB2+IRLM)	0.003379	0.07	#OCCURRENCES : 3967
NONNESTED	52.522821	0.022607	N/A	IRLM LOCK+LATCH	N/A	N/A	#ALLIEDS : 3587
STORED PROC	0.000000	0.000000	N/A	DB2 LATCH	N/A	N/A	#ALLIEDS DISTRIB: 0
UDF	0.000000	0.000000	N/A	SYNCHRON. I/O	0.005358	0.88	#DBATS : 380
TRIGGER	0.000000	0.000000	N/A	DATABASE I/O	0.002200	0.46	#DBATS DISTRIB. : 0
				LOG WRITE I/O	0.003158	0.42	#NO PROGRAM DATA: 0
:							

Note: The number of ALLIED, ALLIED DISTRIBUTED, DBAT, and DBAT DISTRIBUTED threads are presented in the **Highlights** block.

Specifying intervals for long-term reporting

If you want to perform a trend analysis, you probably want to report data by interval. To do this, first reduce the input data for your Accounting reports using an interval other than the default zero. Then, order the report by intervals.

For example, you might want to report the data at daily intervals. In this case you would specify INTERVAL(1440) in the REDUCE subcommand and ORDER(INTERVAL) in the REPORT subcommand. If you do not intend to produce reports at time intervals, use the default (0) for INTERVAL, which means that no interval processing is performed.

Specifying exception thresholds for specific fields

You can set exception thresholds for virtually all Accounting fields. However, carefully consider the fields for which to specify exception thresholds. The more fields you specify, the greater the effects on processing.

Specify exception thresholds only for those fields that you believe will signal poor performance in your environment.

You can also use the TOP (ONLY) subcommand option to identify problems instead of specifying exception thresholds. TOP processing is more economical than exception processing.

Using a DPMOUT data set

Keep the content of the DPMOUT data set for further reports.

After OMEGAMON XE for DB2 PE has generated all reports that you requested, the preprocessed input data is either discarded or written to the output data set DPMOUT, depending on what you specified. The DPMOUT data set can be used as input to OMEGAMON XE for DB2 PE, so if you want to produce reports from the same data in several executions and your SMF/GTF data set is large, you might want to keep the DPMOUT data set.

If you do not specify DPMOUTDD, only the records required for the current job step are processed, which improves performance. If you want to keep a DPMOUT data set for the specific purpose of producing future Accounting reports and traces, specify GLOBAL INCLUDE IFCID(3 239) in the command stream within the JCL. This reduces the size of the DPMOUT data set.

Saving reduced data

This section shows alternative ways to save reduced data. Use the subcommands REDUCE and SAVE (without CONVERT), for example, to store historical Accounting data in smaller data sets.

You can reduce and save data as follows:

- If you want to keep historical Accounting data about DB2 performance in a VSAM data set, consider using REDUCE and SAVE as follows:
 1. REDUCE trace data
 2. SAVE reduced data

Reducing and saving data uses a considerable amount of system resources, but the resulting VSAM-Save data set is much smaller than the original input data set.

The size of the data set depends on the reduction interval that you specified and the type of environment the data is from (for example, the number of different users and plans present in the input data), but it is always much smaller than the original input data set.

You can produce reports from the VSAM data by using the RESTORE command. Remember that you cannot produce traces from reduced data.

You can then proceed as follows:

1. CONVERT saved data: The Save-File utility changes Save data sets into sequential data sets for use by the utility.
 2. LOAD converted data to Performance database (PDB) tables.
- The following steps are required to load Statistics or Accounting Save data using the CONVERT option of the SAVE subcommand:
 1. REDUCE trace data.
 2. CONVERT reduced data using the SAVE subcommand with the CONVERT option. The reduced data is saved and converted in one step into a sequential data set that can be loaded into DB2 tables.
 3. LOAD converted data to Performance database (PDB) tables.

Exception processing

Use exception processing to identify Accounting report, trace, and File data set entries with fields that contain values outside thresholds specified in the Exception Threshold data set. When exception processing is active, the data to be reported is checked against these thresholds. You can obtain exception reports, traces, and File data sets by using the EXCEPTION subcommand option in ACCOUNTING subcommands.

There are two threshold types that you can set: *warning* and *problem*. A warning message is printed if a value is outside the first threshold. A problem message is printed if a value is outside the second threshold.

Accounting exception reports, traces, and File data sets are identical to the usual Accounting reports, traces, and File data sets, except that they only contain entries that have at least one field in exception status. For reports and traces, a block of data that shows the fields in exception status is printed next to the report or trace entry.

Exception logs contain Accounting and Statistics fields that are in exception status in a single report in timestamp order.

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

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

Accounting Exception Messages block

As the following example shows, if EXCEPTION was specified in the REPORT or TRACE subcommand, the entry is formatted and printed in the requested layout, followed by the Exception Messages block. The Exception Messages block identifies the fields in exception status.

```
*****
* TYPE      FIELD ID  FIELD DESCRIPTION          BY      VALUE  THRESHOLD  *
*          *          *          *          *          *          *
* PROBLEM    QXINCRB  INCREMENTAL BINDS        TOTAL      1 > 0      *
*          *          *          *          *          *          *
* PROBLEM    ASCDML   TOTAL SQL DML STATEMENTS  COMMIT    127.25 > 100 *
*          *          *          *          *          *          *
* WARNING    QBACGET  GETPAGES          THREAD    2589.00 > 2500 *
*          *          *          *          *          *          *
*****
```

The following columns are presented in the Exception Messages block:

TYPE The type of exception. Valid values are WARNING and PROBLEM.

FIELD ID

The name of the field from the Exception Threshold data set.

FIELD DESCRIPTION

The description of the field.

FIELD QUALIFIER

The qualifier of the field. It can be blank (for the majority of fields), a buffer pool ID, group buffer pool ID, package or DBRM name, or a remote location name.

BY The basis for the comparison. The following information can be printed in this column:

TOTAL

The threshold is specified as a “total” value.

MINUTE

The threshold is specified as a “by minute” value. The value in the report or trace entry is divided by the number of minutes (class 1 elapsed time) before making the comparison.

SECOND

The threshold is specified as a “by second” value. The value in the report or trace entry is divided by the number of seconds (class 1 elapsed time) before making the comparison.

COMMIT

The threshold is specified as a “by commit” value. The value in the report or trace entry is divided by the number of Commits before making the comparison.

THREAD

The threshold is specified as a “by thread” value. The value in the report or trace entry is divided by the number of threads before the comparison. For traces it is equivalent to *by total*.

VALUE

The actual field value that is used for the comparison. For “by minute”,

“by second”, “by commit”, or “by thread” comparisons, the value that you get after the division is printed. The greater than symbol (>) or the less than symbol (<) is printed between this column and the THRESHOLD column to indicate whether the value is larger or smaller than the threshold value.

THRESHOLD

The threshold defined in the Exception Threshold data set.

ORDER processing

You can change the way reports are summarized by using the ORDER subcommand option of the REPORT subcommand. The ORDER subcommand option specifies which OMEGAMON XE for DB2 PE identifiers are used to aggregate records. It also identifies the presentation sequence of the report entries, unless the TOP(ONLY) subcommand option is specified for Accounting.

You can:

- Specify one entry of ORDER for each REPORT subcommand.
- Order by one, two, or three identifiers separated by a dash.
- Specify up to five sets of the identifiers separated by at least one blank, a comma, or a new line.

The default for ORDER is PRMAUTH-PLANNAME.

In addition to these OMEGAMON XE for DB2 PE identifiers, you can use the REDUCE INTERVAL to order data on Accounting and Statistics reports. See “Example of ordering by interval: ORDER (INTERVAL)” on page 170 for an example of a report ordered by interval.

The following Accounting report examples show ORDER processing for the same input data ordered in different ways:

- “Default ordering: ORDER (PRMAUTH-PLANNAME)”
- “Ordering by connection ID: ORDER (CONNECT)” on page 161
- “Ordering by correlation ID: ORDER (CORRNAME)” on page 162

“Default ordering: ORDER (PRMAUTH-PLANNAME)”

The following example shows a report with default ordering of plan names within primary authorization IDs.

“Ordering by connection ID: ORDER (CONNECT)” on page 161

You can order the report by the connection ID using the subcommand option ORDER (CONNECT).

“Ordering by correlation ID: ORDER (CORRNAME)” on page 162

You can also identify the task by using correlation data:

“Examples of ordering by Plan, Main Package, and Package” on page 163

These examples show data ordered by PLANNAME, MAINPACK, and PACKAGE identifiers.

“Example of ordering by activity name: ORDER (ACTNAME)” on page 166

This example shows how to order an Accounting report by activity name.

“Example of ordering by interval: ORDER (INTERVAL)” on page 170

This example shows a short Accounting report that is ordered by interval.

Default ordering: ORDER (PRMAUTH-PLANNAME)

The following example shows a report with default ordering of plan names within primary authorization IDs.

Accounting report - default order

In this example, data is ordered according to various authorization IDs and plans. All primary authorization IDs and plans present in the input data are shown.

The TOTAL rows are printed for primary authorization IDs that contain more than one plan.

LOCATION: DSNCAT GROUP: DSNCAT MEMBER: SSDQ SUBSYSTEM: SSDQ DB2 VERSION: V11			OMEGAMON XE for DB2 Performance Expert (V5R3M0) Accounting REPORT - SHORT						PAGE: 1-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 07/14/13 18:47:13.28 TO: 07/14/13 19:55:28.69				
ORDER: PRIMAUTH-PLANNAME SCOPE: MEMBER													
PRIMAUTH PLANNAME	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK SUS #LOCKOUT

ADMF001 DDL3P04	2 0	7 16	1.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	12:22.510536 0.759452			N/P N/P	44.00 0.00	9.50 0.00	0.00 0
ADMF001 DSNUTIL	10 0	2 87	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	3:05.083686 1.899589			N/P N/P	625.50 315.10	19.40 7.60	0.10 0
ADMF001 QMF311	10 0	10 0	0.00 1.00	0.00 1.00	0.00 0.00	0.00 1.00	1:06.989512 0.041995			N/P N/P	23.00 0.00	0.20 0.00	3.00 10
*** TOTAL ***													
ADMF001	22 0	19 103	0.09 0.45	0.00 0.45	0.00 0.00	0.00 0.45	3:02.079684 0.951579			N/P N/P	298.77 143.23	9.77 3.45	1.41 10
ADMF002 DDL3P08	2 0	159 4	1.50 170.00	0.00 1.50	81.00 0.50	0.00 0.00	1:54:45.002759 8.915043			N/P N/P	2589.00 362.50	284.50 121.00	207.50 156
ADMF003 DDL3P16	2 0	34 1	25.50 0.00	0.00 0.00	0.00 0.00	1.00 0.00	1:54:00.564496 6.607316			N/P N/P	1459.50 77.00	93.00 54.50	82.00 32
ADMF005 DSNUTIL	6 0	0 36	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	1:26.881992 0.119407			N/P N/P	54.67 15.83	2.50 2.50	1.00 0
ADMF006 DSNTEP31	1 0	0 1	0.00 76.00	0.00 3.00	0.00 3.00	0.00 3.00	54.526350 0.326120			N/P N/P	587.00 8.00	46.00 17.00	3.00 0
ADMF006 DSNUTIL	2 0	0 16	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	56.913775 0.447878			N/P N/P	53.50 8.00	0.50 2.50	1.50 0
*** TOTAL ***													
ADMF006	3 0	0 17	0.00 25.33	0.00 1.00	0.00 1.00	0.00 1.00	56.117967 0.407292			N/P N/P	231.33 8.00	15.67 7.33	2.00 0
*** GRAND TOTAL ***													
	35 0	212 161	1.60 12.17	0.00 0.46	4.63 0.11	0.06 0.37	15:18.472383 1.540508			N/P N/P	448.34 118.54	29.49 13.26	17.77 198

Accounting REPORT COMPLETE

Ordering by connection ID: ORDER (CONNECT)

You can order the report by the connection ID using the subcommand option ORDER (CONNECT).

Here is a sample of how to specify the command:

```
:  
:  
ACCOUNTING  
REPORT  
ORDER (CONNECT)  
:  
:
```

Accounting report - ordered by connection ID

The following example shows a report with ordering by connection ID.

Three connection IDs are reported: information for connection ID BATCH is reported on the first two lines followed by information for connections DB2CALL and UTILITY.

GRAND TOTAL is printed at the end of the report. It shows the aggregated values for all three connection IDs.

LOCATION: DSNCAT GROUP: DSNCAT MEMBER: SSDQ SUBSYSTEM: SSDQ DB2 VERSION: V11			OMEGAMON XE for DB2 Performance Expert (V5R3M0) Accounting REPORT - SHORT					PAGE: 1-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 07/14/13 18:47:13.28 TO: 07/14/13 19:55:28.69				
			ORDER: CONNECT SCOPE: MEMBER									
CONNECT	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1 CPUTIME	EL.TIME CLASS2 CPUTIME	CLASS2 CLASS2 CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK SUS #LOCKOUT
BATCH	7 0	200 22	8.00 59.43	0.00 0.86	23.14 0.57	0.29 0.43	1:09:01.525990 4.698535	N/P N/P	1253.14 126.71	117.14 52.57	83.14 188	
DB2CALL	10 0	10 0	0.00 1.00	0.00 1.00	0.00 0.00	0.00 1.00	1:06.989512 0.041995	N/P N/P	23.00 0.00	0.20 0.00	3.00 10	
UTILITY	18 0	2 139	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	2:18.108687 1.144894	N/P N/P	371.67 181.22	11.67 5.33	0.56 0	
*** GRAND TOTAL ***	35 0	212 161	1.60 12.17	0.00 0.46	4.63 0.11	0.06 0.37	15:18.472383 1.540508	N/P N/P	448.34 118.54	29.49 13.26	17.77 198	

Accounting REPORT COMPLETE

Ordering by correlation ID: ORDER (CORRNAME)

You can also identify the task by using correlation data:

```

:
ACCOUNTING
REPORT
ORDER (CORRNAME)
:

```

Note: In a distributed environment reports should be ordered by REQLOC or CONNTYPE. If REQLOC or CONNTYPE are not used in the ORDER subcommand option of REPORT, the Accounting portion of *all* threads (including DBATs) where the combination of OMEGAMON XE for DB2 PE identifiers is the same is reported as one entry.

Accounting report - ordered by correlation ID

The following example shows an Accounting report ordered by correlation ID.

LOCATION: DSNCAT GROUP: DSNCAT MEMBER: SSDQ SUBSYSTEM: SSDQ DB2 VERSION: V11			OMEGAMON XE for DB2 Performance Expert (V5R3M0) Accounting REPORT - SHORT					PAGE: 1-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 07/14/13 18:47:13.28 TO: 07/14/13 19:55:28.69				
			ORDER: CORRNAME SCOPE: MEMBER									
CORRNAME	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1 CPUTIME	EL.TIME CLASS2 CPUTIME	CLASS2 CLASS2 CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK SUS #LOCKOUT
RECOVER0	10 0	2 87	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	3:05.083686 1.899589	N/P N/P	625.50 315.10	19.40 7.60	0.10 0	
RUNPRG16	2 0	34 1	25.50 0.00	0.00 0.00	0.00 0.00	1.00 0.00	1:54:00.564496 6.607316	N/P N/P	1459.50 77.00	93.00 54.50	82.00 32	
RUNPRG4	2 0	7 16	1.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	12:22.510536 0.759452	N/P N/P	44.00 0.00	9.50 0.00	0.00 0	
RUNPRG8	2 0	159 4	1.50 170.00	0.00 1.50	81.00 0.50	0.00 0.00	1:54:45.002759 8.915043	N/P N/P	2589.00 362.50	284.50 121.00	207.50 156	
RUNUTL10	2 0	0 12	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	1:43.715927 0.081900	N/P N/P	38.00 17.00	0.50 0.50	1.00 0	
RUNUTL11	2 0	0 12	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	1:57.605935 0.136053	N/P N/P	62.50 15.00	0.50 4.00	1.00 0	
RUNUTL12	3 0	0 17	0.00 25.33	0.00 1.00	0.00 1.00	0.00 1.00	56.117967 0.407292	N/P N/P	231.33 8.00	15.67 7.33	2.00 0	
RUNUTL13	2 0	0 0	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	39.324115	N/P	63.50	6.50	1.00	

	0	12	0.00	0.00	0.00	0.00	0.140269	N/P	15.50	3.00	0
SYSADM	10	10	0.00	0.00	0.00	0.00	1:06.989512	N/P	23.00	0.20	3.00
	0	0	1.00	1.00	0.00	1.00	0.041995	N/P	0.00	0.00	10
*** GRAND TOTAL ***	35	212	1.60	0.00	4.63	0.06	15:18.472383	N/P	448.34	29.49	17.77
	0	161	12.17	0.46	0.11	0.37	1.540508	N/P	118.54	13.26	198

ACCOUNTING REPORT COMPLETE

Examples of ordering by Plan, Main Package, and Package

These examples show data ordered by PLANNAME, MAINPACK, and PACKAGE identifiers.

The data shown in “Accounting trace - input data for ordering reports” is used to produce the reports shown in “Accounting report - ordered by plan” on page 164, “Accounting report - ordered by plan and MAINPACK - sample” on page 165, and “Accounting report - ordered by package - sample” on page 166.

Accounting trace - input data for ordering reports

The data in the following example has been simplified to ease readability.

LOCATION: STLEC1	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 7-1
GROUP: N/P	Accounting TRACE - SHORT	REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P		TO: NOT SPECIFIED
SUBSYSTEM: V51A		ACTUAL FROM: 08/25/13 22:02:59.55
DB2 VERSION: V10		PAGE DATE: 08/25/13

PRIMAUTH	CORRNAME	CONNECT	ACCT	TIMESTAMP	COMMITTS	OPENS	UPDATES	INSERTS	EL. TIME(CL1)	EL. TIME(CL2)	GETPAGES	SYN.READ	LOCK	SUS
PLANNAME	CORRNMBR	THR.TYPE	TERM.	CONDITION	SELECTS	FETCHES	DELETES	PREPARE	CPU TIME(CL1)	CPU TIME(CL2)	BUF.UPDT	TOT.PREF	LOCKOUTS	
XXUSER01	CORRXXXX	BATCH	12:00:00.000000		1	1	0	0	10.000000	4.444444	2	0	0	
PLANX	'BLANK'	ALLIED	NORM DEALLOC		0	1	0	0	0.100000	0.044444	2	0	0	

PROGRAM NAME	TYPE	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP
PACKA	PACKAGE	1	1.000000	0.010000	0.000000	0				
PACKB	PACKAGE	1	2.000000	0.020000	0.000000	0				
PACKC	PACKAGE	1	1.000000	0.010000	0.000000	0				

XXUSER01	CORRXXXX	BATCH	12:01:00.000000		1	1	0	0	20.000000	15.555555	1	0	0	
PLANX	'BLANK'	ALLIED	NORM DEALLOC		0	1	0	0	0.200000	0.155555	1	0	0	

PROGRAM NAME	TYPE	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP
PACKB	PACKAGE	1	10.000000	0.100000	0.000000	0				
PACKD	PACKAGE	1	4.000000	0.040000	0.000000	0				

XXUSER01	CORRXXXX	BATCH	12:02:00.000000		1	1	0	0	30.000000	16.666666	4	0	0	
PLANY	'BLANK'	ALLIED	NORM DEALLOC		0	1	0	0	0.300000	0.166666	0	0	0	

PROGRAM NAME	TYPE	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP
PACKD	PACKAGE	1	10.000000	0.100000	0.000000	0				
PACKA	PACKAGE	1	5.000000	0.050000	0.000000	0				

XXUSER01	CORRXXXX	BATCH	12:03:00.000000		1	1	0	0	30.000000	8.888888	3	0	0	
PLANX	'BLANK'	ALLIED	NORM DEALLOC		0	1	0	0	0.300000	0.088888	0	0	0	

PROGRAM NAME	TYPE	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP
PACKB	PACKAGE	1	6.000000	0.060000	0.000000	0				
PACKD	PACKAGE	1	2.000000	0.020000	0.000000	0				

The input data contains information about:

- Two plans. PLANX is used by three threads and PLANY by one thread. The first instance of PLANX shows that three packages (PACKA, PACKB, and PACKC) are executed. The elapsed time for the entire PLANX is 10 seconds and CPU times for the individual packages are 0.01, 0.02, and 0.01 seconds.

- Four packages. Different combinations of packages were executed under the different plans, because a plan does not necessarily invoke the same packages each time it is executed. This can happen when, for example, a number of packages are bound in a single plan in a CICS environment and different packages are executed in different circumstances.

Ordering by Plan

This example shows the result of ordering the input data by plan. The following command was used to produce the example shown in “Accounting report - ordered by plan.”

```

:
:
ACCOUNTING
REPORT
ORDER (PLANNAME)
:
:

```

Accounting report - ordered by plan

The following example shows an Accounting report that contains an entry for both plans in the input data.

Data for the different packages is summarized under the plans. Also, different DB2 executions of PLANX are summarized in one entry.

LOCATION: DSNCAT		OMEGAMON XE for DB2 Performance Expert (V5R3M0)						PAGE: 1-1							
GROUP: DSNCAT		Accounting REPORT - SHORT						REQUESTED FROM: NOT SPECIFIED							
MEMBER: SSDQ								TO: NOT SPECIFIED							
SUBSYSTEM: SSDQ		ORDER: PLANNAME						INTERVAL FROM: 07/14/13 18:47:13.28							
DB2 VERSION: V10		SCOPE: MEMBER						TO: 07/14/13 19:55:28.69							
PLANNAME		#OCCURS	#ROLLBK	SELECTS	INSERTS	UPDATES	DELETES	CLASS1	EL.TIME	CLASS2	EL.TIME	GETPAGES	SYN.READ	LOCK	SUS
		#DISTR	#COMMIT	FETCHES	OPENS	CLOSES	PREPARE	CLASS1	CPUTIME	CLASS2	CPUTIME	BUF.UPDT	TOT.PREF	#LOCKOUT	

PLANX		3	0	0.00	0.00	0.00	0.00		20.000000		9.629629	2.00	0.00	0.00	
		0	3	1.00	1.00	1.00	0.00		0.200000		0.096292	1.00	0.00	0	

PROGRAM NAME		TYPE	#OCCURS	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP			
PACKA		PACKAGE	1	1.00		1.000000		0.010000		0.000000		0.00			
PACKB		PACKAGE	3	1.00		6.000000		0.060000		0.000000		0.00			
PACKC		PACKAGE	1	1.00		1.000000		0.010000		0.000000		0.00			
PACKD		PACKAGE	2	1.00		3.000000		0.030000		0.000000		0.00			

PLANY		1	0	0.00	0.00	0.00	0.00		30.000000		16.666667	4.00	0.00	0.00	
		0	1	1.00	1.00	1.00	0.00		0.300000		0.166667	0.00	0.00	0	

PROGRAM NAME		TYPE	#OCCURS	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP			
PACKD		PACKAGE	1	1.00		10.000000		0.100000		0.000000		0.00			
PACKA		PACKAGE	1	1.00		5.000000		0.050000		0.000000		0.00			

Ordering by Plan and MAINPACK: ORDER (PLANNAME-MAINPACK)

The MAINPACK identifier is used to distinguish between records with the same plan name, but which executed different packages.

This example shows the result of ordering the input data by plan and MAINPACK.

MAINPACK identifies a representative package within the plan. The first package ID is the default for MAINPACK. However, in this example, the MAINPACK member of the DPMPARMS data set has been modified so that it is the package ID of the last executed package, see “Defining the MAINPACK identifier” on page 252 for information about MAINPACK.

In the input data shown in “Accounting trace - input data for ordering reports” on page 163, it is assumed that PACKC is the last executed package in the first entry for PLANX, PACKD for the second and third entries of PLANX, and PACKA for the entry of PLANY.

The following command was used to produce the report in “Accounting report - ordered by plan and MAINPACK - sample.”

```

:
:
ACCOUNTING
REPORT
ORDER (PLANNAME-MAINPACK)
:
:

```

Accounting report - ordered by plan and MAINPACK - sample

The following example shows an Accounting report that contains an entry for each combination of PLANNAME and MAINPACK.

LOCATION: DSNCAT		OMEGAMON XE for DB2 Performance Expert (V5R3M0)							PAGE: 1-1						
GROUP: DSNCAT		Accounting REPORT - SHORT							REQUESTED FROM: NOT SPECIFIED						
MEMBER: SSDQ									TO: NOT SPECIFIED						
SUBSYSTEM: SSDQ		ORDER: PLANNAME-MAINPACK							INTERVAL FROM: 07/14/13 18:47:13.28						
DB2 VERSION: V11		SCOPE: MEMBER							TO: 07/14/13 19:55:28.69						
PLANNAME		#OCCURS	#ROLLBK	SELECTS	INSERTS	UPDATES	DELETES	CLASS1	EL.TIME	CLASS2	EL.TIME	GETPAGES	SYN.READ	LOCK	SUS
MAINPACK		#DISTR	#COMMIT	FETCHES	OPENS	CLOSES	PREPARE	CLASS1	CPUTIME	CLASS2	CPUTIME	BUF.UPDT	TOT.PREF	#LOCKOUT	

PLANX			1	0	0.00	0.00	0.00		10.000000		4.444444	2.00	0.00	0.00	
PACKC			0	1	1.00	1.00	0.00	0.00	0.100000		0.044444	2.00	0.00	0	

PROGRAM NAME		TYPE	#OCCURS	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP			
PACKA		PACKAGE	1	1.00		1.000000		0.010000		0.000000		0.00			
PACKB		PACKAGE	1	1.00		2.000000		0.020000		0.000000		0.00			
PACKC		PACKAGE	1	1.00		1.000000		0.010000		0.000000		0.00			

PLANX			2	0	0.00	0.00	0.00	0.00	25.000000		12.222222	2.00	0.00	0.00	
PACKD			0	1	1.00	1.00	1.00	0.00	0.250000		0.122222	0.50	0.00	0	

PROGRAM NAME		TYPE	#OCCURS	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP			
PACKB		PACKAGE	2	1.00		8.000000		0.080000		0.000000		0.00			
PACKD		PACKAGE	2	1.00		3.000000		0.030000		0.000000		0.00			

PLANY			1	0	0.00	0.00	0.00	0.00	30.000000		16.666666	4.00	0.00	0.00	
PACKA			0	1	1.00	1.00	1.00	0.00	0.300000		0.166666	0.00	0.00	0	

PROGRAM NAME		TYPE	#OCCURS	SQLSTMT	CL7	ELAP.TIME	CL7	CPU TIME	CL8	SUSP.TIME	CL8	SUSP			
PACKD		PACKAGE	1	1.00		10.000000		0.100000		0.000000		0.00			
PACKA		PACKAGE	1	1.00		5.000000		0.050000		0.000000		1.00			

There are two entries for PLANX:

- One entry where PACKC is the last package executed.
- Another entry where PACKD is the last package executed. Threads 2 and 4 are combined in this entry.

There is one entry for PLANY with the representative package PACKA.

You cannot use this report to attribute the General Accounting data to one package, unless only one package exists within a plan.

Ordering by Package or DBRM: ORDER (PACKAGE)

The previous examples present packages within plans. To summarize the package Accounting data regardless of the plan under which the packages or DBRMs were executed, you can order by package.

This example shows the result of ordering the input data by package.

The following command was used to produce the report in “Accounting report - ordered by package - sample.”

```

:
:
ACCOUNTING
REPORT
ORDER (PACKAGE)
:
:

```

Accounting report - ordered by package - sample

The following example shows the use of resources on a per package/DBRM basis in an Accounting report, regardless of the plan under which a particular package is executed.

LOCATION: DSNCAT	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 1-1
GROUP: DSNCAT	Accounting REPORT - SHORT	REQUESTED FROM: NOT SPECIFIED
MEMBER: SSDQ		TO: NOT SPECIFIED
SUBSYSTEM: SSDQ	ORDER: PACKAGE	INTERVAL FROM: 07/14/13 18:47:13.28
DB2 VERSION: V11	SCOPE: MEMBER	TO: 07/14/13 19:55:28.69

PACKAGE	TYPE	#OCCURS	SQLSTMT	CL7.CPU TIME	CL8 SUSP
			CL7 ELAP.TIME	CL8 SUSP.TIME	
APC1.COLL1.PACKA	PACKAGE	2	1.00 3.000000	0.030000 0.000000	0.00
APC1.COLL1.PACKB	PACKAGE	3	1.00 6.000000	0.060000 0.000000	0.00
APC1.COLL1.PACKC	PACKAGE	1	1.00 1.000000	0.010000 0.000000	0.00
APC1.COLL1.PACKD	PACKAGE	3	1.00 5.333333	0.053333 0.000000	0.00

Note: Accounting reports that are ordered by package identifier (created by using the PACKAGE keyword with the ORDER subcommand option) show only the following report blocks:

- Package Identification
- Times - Class 7 - Package Times
- Package Suspensions
- Global Contention L-Locks
- Global Contention P-Locks
- Package Buffer Pool Activity
- Package SQL Activity
- Package Locking Activity

Example of ordering by activity name: ORDER (ACTNAME)

This example shows how to order an Accounting report by activity name.

If you want to use ORDER (ACTNAME), you should consider the following:

- ACTNAME cannot be combined with other OMEGAMON XE for DB2 PE identifiers in the ORDER subcommand.
- ORDER (ACTNAME) is only allowed for reports that are created from input trace specified with //INPUTDD.
- If you use the RESTORE subcommand, you cannot order by activity name in the subsequent REPORT subcommands.

The following command orders an Accounting report by activity name.

```

:
:
ACCOUNTING
REPORT

```

ORDER (ACTNAME)
 :

To identify the real number of stored procedures (SP) or user-defined functions (UDF) instead of the number of packages used, you must include the trace collection of IFCID 233 (Start/Stop for each SP or UDF), 380 (Start/Stop for SP), or 381 (Start/Stop for UDF) in your input trace data. Stored procedures and user-defined functions are determined and can be counted if IFCID 233, 380, or 381 is found. Subprograms called by these routines and functions are not taken into account for the number of occurrences.

Note: If there is neither IFCID 233, 380, nor 381 for each SP or UDF, all packages of the same activity type and activity name are counted as separate occurrences. To calculate the average value, the sum is divided by the number of:

- Occurrences (prior to DB2 10).
- Threads to roll data into this QPAC data section (DB2 10 or later).

Sample workload with two different SPs and subprograms

Table 17 shows a workload of six threads with stored procedures (SP) and subprograms. In this sample workload IFCID 233 is written for each SP execution. Package SP-A has the activity name ACTNAME-A. Package SP-B has the activity name ACTNAME-B. Packages PKG1, PKG2, or PKG3 are linked by SP-A or SP-B that is why they inherit their activity names.

Table 17. Sample workload with two different SPs and six executions in total

Six threads (plans)	with SP and subprograms	DB2 Trace records (3, 239) written				SP START/STOP IFCID 233	
Plan A	SP-A links PKG1 links PKG2	(3) Plan A	(239) SP-A	(239) PKG1	(239) PKG2	(233) Start	(233) Stop
Plan A	SP-A links PKG1	(3) Plan A	(239) SP-A	(239) PKG1		(233) Start	(233) Stop
Plan B	SP-B	(3) Plan B	(239) SP-B			(233) Start	(233) Stop
Plan A	SP-A links PKG2 links PKG1	(3) Plan A	(239) SP-A	(239) PKG2	(239) PKG1	(233) Start	(233) Stop
Plan B	SP-B links PKG1	(3) Plan B	(239) SP-B	(239) PKG1		(233) Start	(233) Stop
Plan A	SP-A links PKG2 links PKG3	(3) Plan A	(239) SP-A	(239) PKG2	(239) PKG3	(233) Start	(233) Stop

“Comparison of Accounting report types based on the above sample workload” on page 168

The report layout depends on the OMEGAMON XE for DB2 PE subcommand option of ORDER specified for producing an Accounting report.

“The impact of the DB2 subsystem parameter ACCUMACC” on page 169

The DB2 subsystem parameter ACCUMACC determines whether DB2 Accounting data is to be accumulated by the user for DDF and RRSF threads.

Comparison of Accounting report types based on the above sample workload

The report layout depends on the OMEGAMON XE for DB2 PE subcommand option of ORDER specified for producing an Accounting report.

If the DB2 subsystem parameter ACCUMACC is used, the report layout provided with DB2 10 differs from the report layout provided with a DB2 version prior to DB2 10.

Each report is based on the sample workload in Table 17 on page 167.

Subcommand ORDER (PLANNAME)

The following report is a typical Accounting report ordered by various identifiers on plan level with their packages underneath. It is produced if you specify, for example, ORDER (PLANNAME):

Plan A occurrences=4
Pack SP-A occurrences=4
Pack PKG1 occurrences=3
Pack PKG2 occurrences=3
Pack PKG3 occurrences=1
Plan B occurrences=2
Pack SP-B occurrences=2
Pack PKG1 occurrences=1

Subcommand ORDER (PACKAGE)

The following sample shows a report:

- Ordered by package
- Average for each package
- Average cross plan
- Occurrences are equal to the number of package sections found in all plans

The following Accounting report is ordered by package, which reports on package level cross thread/plan usage. It is produced if you specify, for example, ORDER (PACKAGE):

Pack SP-A occurrences=4
Pack PKG1 occurrences=4
Pack PKG2 occurrences=3
Pack PKG3 occurrences=1
Pack SP-B occurrences=2

Subcommand ORDER (ACTNAME) with IFCID 233 or 380 available for each stored procedure

The following sample shows a report:

- Ordered by activity name
- Average for each SP
- Includes all processing subprograms under SP
- Occurrences are equal to the **number of SP calls**

The following Accounting report is ordered by activity name (stored procedure) where all activities (subprograms) underneath the stored procedure are included in the SP average values. It is produced if you specify ORDER (ACTNAME) and your input trace data includes IFCID 233 or 380 for each SP:

```
Activity Name ACTNAME_A occurrences=4
Activity Name ACTNAME_B occurrences=2
```

Note: Explanation for the above sample:

- Under SP-A with the activity name ACTNAME_A the **average execution of SP-A** is shown which includes all PKG1, PKG2, and PKG3 workloads associated with this stored procedure.
- Under SP-B with the activity name ACTNAME_B the **average execution of SP-B** is shown which includes all PKG1 workloads associated with this stored procedure.

Subcommand ORDER (ACTNAME) without IFCID 233 or 380 in a DB2 trace

The following sample shows a report:

- Ordered by activity name
- Average for each SP
- Includes all processing under SP
- Occurrences are equal to the **number of package sections**, including SP as a package

The following Accounting report is ordered by activity name (stored procedure) where all activities (subprograms) underneath the SP are included. The average values shown represent the aggregation of all packages including SPs executed within an SP. However, all subprograms are counted as separate occurrences because no IFCID 233 or 380 were found by the reporting program:

```
Activity Name Pack ACTNAME_A occurrences=11
Activity Name Pack ACTNAME_B occurrences=3
```

Note: Explanation for the above sample:

- Under SP-A with the activity name ACTNAME_A the **average of the aggregation for each package section** executed under SP-A (ACTNAME_A, PKG1, PKG2, PKG3) is shown.
- Under SP-B with the activity name ACTNAME_B the **average of the aggregation for each package section** executed under SP-B (ACTNAME_B and PKG1) is shown.

The impact of the DB2 subsystem parameter ACCUMACC

The DB2 subsystem parameter ACCUMACC determines whether DB2 Accounting data is to be accumulated by the user for DDF and RRSF threads.

If ACCUMACC is greater than 0, the threads which relate to the same Accounting accumulation identifiers (defined by the aggregation field ACCUMUID) are aggregated into one Accounting trace record (3, 239).

For DB2 versions prior to version 10 all packages of these accumulated threads are reported into **one** package section. As a result you lose the granularity for each package and you cannot:

- Distinguish between SP and called subprograms.
- Determine which thread has executed which SP or package.

Starting with DB2 10 reports show the first 24 different packages or SPs of the accumulated threads in separate data sections (239) and only if there are more than 24 packages, all the remaining packages will be accumulated into the 25th data section.

For more information on DB2 subsystem parameters refer to the *IBM Knowledge Center*.

Example of ordering by interval: ORDER (INTERVAL)

This example shows a short Accounting report that is ordered by interval.

To produce a report that presents DB2 activity by time intervals, the input data first needs to be reduced to the intervals that you want to use in reporting.

The following command was used to generate the sample report in “Accounting report - ordered by interval.”

```

:
:
ACCOUNTING
  REDUCE
    INTERVAL (5)
  REPORT
    ORDER (INTERVAL)
:
:

```

In this command the data is reduced to 5-minute intervals. The report is ordered by this interval. An entry is produced that shows the activity during every 5-minute interval.

Accounting report - ordered by interval

The following example shows an Accounting report that is ordered by interval. Reports ordered by INTERVAL, with or without another identifier, are especially useful in trend analysis. For example, you can reduce and save your data by specifying INTERVAL(1440) to produce reports that show the day-by-day activity of your DB2 subsystem.

LOCATION: DSNCAT GROUP: DSNCAT MEMBER: SSDQ SUBSYSTEM: SSDQ DB2 VERSION: V11		OMEGAMON XE for DB2 Performance Expert (V5R3M0) Accounting REPORT - SHORT						PAGE: 1-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 07/14/13 18:47:13.28 TO: 07/14/13 20:55:28.69					
		ORDER: INTERVAL SCOPE: MEMBER											
INTERVAL	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1 CPUTIME	EL.TIME CLASS2 CPUTIME	CLASS2 EL.TIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK #LOCKOUT	SUS
05/10 19:50 - 05/10 19:55	1 0	1 3	3.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00	5:23.338584 0.087963	N/P N/P	51.00 0.00	20.00 0.00	1.00 0		
05/10 19:55 - 05/10 20:00	6 0	8 36	11.00 0.00	0.67 0.00	0.00 0.00	0.00 0.00	2:29.264737 4.134166	N/P N/P	231.00 0.67	17.67 1.00	22.00 0		
05/10 20:00 - 05/10 20:05	1 0	1 0	2.00 5.00	0.00 5.00	0.00 5.00	0.00 0.00	12:20.702202 7.753375	N/P N/P	6232.00 2931.00	299.00 8.00	23.00 1		
05/10 20:05 - 05/10 20:10	7 0	10 6	2.57 0.00	0.00 0.00	0.00 0.00	0.00 0.14	10:01.005681 2.235738	N/P N/P	552.57 192.86	26.00 0.29	21.43 5		
*** GRAND TOTAL ***													

15	20	5.93	0.27	0.00	0.00	6:51.111265	N/P	769.13	40.47	20.40
0	45	0.33	0.33	0.33	0.07	3.219767	N/P	285.67	1.07	6

ACCOUNTING REPORT COMPLETE

TOP processing

TOP processing is useful to identify the report entries that might indicate a problem application.

Signs of a problem can be a long elapsed or processing time, a high number of suspensions, commits per update, or buffer updates. Identifying these entries is especially useful when your report covers many users or a long period so that it is not immediately clear which applications are causing performance problems.

To identify report entries with a high value in certain fields, you can produce an Accounting report or trace with TOP lists. TOP lists are index-like reports at the end of a report or trace. They point out the most interesting entries in the report or trace. You can generate such lists by using the TOP subcommand option.

Note: Entries with 0 or undetermined values are not shown. Moreover, if your input data contains only 0 or undetermined values for the TOP fields requested, a TOP list is not generated.

You can also filter a report or trace so that only the main resource consumers are shown, ordered by descending resource value. To obtain a filtered report or trace, specify the TOP subcommand option with the ONLY keyword.

The shown resource values are average values. To obtain total values, which means, the main resource consumers calculated by taking into account how often they consumed resources, specify the TOP subcommand option with the TOTAL keyword. TOTAL only applies to reports. If you specify it with a trace, it is ignored.

“Examples of TOP processing”

These examples show some applications of the TOP subcommand option in Accounting reports and traces.

Examples of TOP processing

These examples show some applications of the TOP subcommand option in Accounting reports and traces.

Top-10 plans in Accounting report

To produce a short Accounting report with a list of the top-10 plans that spent the longest time in an application, specify:

```

:
GLOBAL
  INCLUDE (LOCATION (USIBMSNEWY11))
  EXCLUDE (CONNTYPE (CICS))
ACCOUNTING
  REPORT
    ORDER (PLANNAME)

```

TOP
:

Accounting report - TOP listing - example

The following example shows the last page of an Accounting report. It presents a list of the 10 plans that had the highest value for elapsed time in the application. The TOP list also shows the page on which the report entries can be found.

```
LOCATION: USIBMNEWY11      OMEGAMON XE for DB2 Performance Expert (V5R3M0)      PAGE: 1-1
GROUP: DSNCAT              Accounting REPORT - SHORT              REQUESTED FROM: NOT SPECIFIED
MEMBER: SSDQ                ORDER: PLANNAME                      TO: NOT SPECIFIED
SUBSYSTEM: SSDQ             SCOPE: MEMBER                      INTERVAL FROM: 07/14/13 18:47:13.28
DB2 VERSION: V10                                TO: 07/14/13 19:55:28.69

ELAPSED TIME SPENT IN APPLICATION      TOP NUMBER REQUESTED: 10

-----
PLANNAME                                VALUE                                PAGE
-----
1  PVLDD4C3                             13:36.924222                        1-1
2  PVLDD4B3                             11:25.333551                        1-1
3  PVLDD4D1                             10:19.115316                        1-1
4  PVLDD4A4                             9:18.056476                         1-1
5  DSNBIND                              8:50.415594                         1-1
6  DSNTEP31                             8:14.506780                        1-1
7  PVLDD4D6                             5:23.338584                        1-1
8  PVLDD4A5                             4:06.067683                        1-1
9  PVLDD4C4                             3:10.939686                        1-1
10 PVLDD4D2                             2:34.389670                        1-1

Accounting REPORT COMPLETE
```

Top-5 Getpages in Accounting trace

To produce a trace with a TOP list for the number of Getpage requests, use the following command:

```
:
ACCOUNTING
TRACE
TOP (5 GETPAGES)
:
```

Accounting trace - TOP listing - example

The following example shows the last page of an Accounting trace. The TOP list on the last page of the trace consists of the five entries that had the highest number of Getpage requests. Each entry consists of TIMESTAMP, PRIMAUTH, and PLANNAME information, the value of the number of Getpage requests, and the page number where the trace entry can be found.

```
LOCATION: STLEC1      OMEGAMON XE for DB2 Performance Expert (V5R3M0)      PAGE: 7-1
GROUP: N/P           Accounting TRACE - SHORT              REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P           ACTUAL FROM: 08/25/13 22:02:59.55
SUBSYSTEM: V51A      PAGE DATE: 08/25/13
DB2 VERSION: V10

GETPAGE REQUESTS      TOP NUMBER REQUESTED: 5

-----
TIMESTAMP
PRIMAUTH
PLANNAME                                VALUE                                PAGE
-----
03/10/13 20:04:20.248358
SYSADM
1  PVLDD4B3                             6232                                1-1
03/10/13 20:06:02.809223
SYSADM
2  PVLDD4C3                             2422                                1-1
03/10/13 20:05:27.859210
SYSADM
3  PVLDD4D1                             675                                 1-1
03/10/13 19:55:02.046545
SYSADM
4  PVLDD4D2                             584                                 1-1
03/10/13 20:05:17.508593
```


TOP lists of available fields in Accounting report

You can generate Accounting reports and traces that show TOP lists for all fields available for use with the TOP subcommand option, as shown in the following command:

```

:
:
ACCOUNTING
  REPORT
    TOP (*)
:
:

```

Top-3 default field in Accounting trace

To produce a filtered trace that shows only the top-3 entries for the default TOP field, namely ELAPSED TIME IN APPLICATION, use the following command:

```

:
:
Accounting
  TRACE
    TOP (3 ONLY)
:
:

```

Accounting trace - TOP ONLY list - example

The following example shows an Accounting trace with a TOP ONLY list.

Note: The TOP(ONLY) subcommand option changes the presentation sequence of Accounting reports and traces. Report and trace entries are ordered according to the TOP resource instead of the timestamp or ORDER subcommand options. The summarization in reports is not affected by the changed sequence caused by TOP filtering.

```

LOCATION: STLEC1          OMEGAMON XE for DB2 Performance Expert (V5R3M0)          PAGE: 7-1
GROUP: N/P              Accounting TRACE - SHORT                                REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P                                                     TO: NOT SPECIFIED
SUBSYSTEM: V51A          ACTUAL FROM: 08/25/13 22:02:59.55
DB2 VERSION: V10        PAGE DATE: 08/25/13

ELAPSED TIME SPENT IN APPLICATION          TOP NUMBER REQUESTED: 3

PRIMAUTH CORRNAME CONNECT ACCT TIMESTAMP COMMITS OPENS UPDATES INSERTS EL. TIME(CL1) EL. TIME(CL2) GETPAGES SYN.READ LOCK SUS
PLANNAME CORRNMBR THR.TYPE TERM. CONDITION SELECTS FETCHES DELETES PREPARE TCB TIME(CL1) TCB TIME(CL2) BUF.UPDT TOT.PREF LOCKOUTS
-----
ALAN      ALAN      DB2CALL 06:10:27.947430      1      0      0      0      7.875259      7.866377      20      0      1
RXSQL     'BLANK'    ALLIED  NORM DEALLOC        0      0      1      0      0.023011      0.017931      6      1      0
ALAN      ALAN      DB2CALL 06:09:06.509263      1      0      0      0      4.175839      4.167954      20      0      1
RXSQL     'BLANK'    ALLIED  NORM DEALLOC        0      0      1      0      0.020973      0.016198      6      1      0
ALAN      ALAN      DB2CALL 06:07:59.399773      1      0      0      1      2.698825      2.689779      15      0      1
RXSQL     'BLANK'    ALLIED  NORM DEALLOC        0      0      0      0      0.017036      0.012228      2      0      0

Accounting TRACE COMPLETE

```

Reducing data

Use the REDUCE subcommand to consolidate records with certain common characteristics into a single record.

You can limit the range of records by date and time. You can specify multiple ranges of time. This can be useful for monitoring peak-time performance, for example.

Within a specified time range, you can specify the interval at which records are consolidated.

The start time of the first interval that is processed by REDUCE is influenced by BOUNDARY, INTERVAL, and FROM.

OMEGAMON XE for DB2 PE attempts to reduce all data that falls between FROM and TO dates and times. The first interval starts at a time aligned with BOUNDARY, at or before the FROM time. If an interval cannot be aligned with the FROM time, the first properly aligned interval starting before the FROM time is used.

Although there is no restriction on the INTERVAL and BOUNDARY combination, your specification should comply with the following rules:

- For intervals of less than 60 (excluding 0), there should be a whole number of intervals in an hour. Choose one of the following values:
 - 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30
- For intervals of 60 or greater, there should be a whole number of intervals in a day. Choose one of the following values:
 - 60, 120, 180, 240, 360, 480, 720, or 1440
- For intervals of one day (1440) or greater, INTERVAL should be a multiple of 1440.
- Select your interval and boundary so that the first interval starts at the FROM time.

“Examples of interval calculation”

These examples show how the REDUCE subcommand is used in combination with FROM, TO, INTERVAL, and BOUNDARY to align start times and intervals.

“Processing intervals” on page 175

Interval processing within the REDUCE subcommand determines the time intervals that are used for reducing Accounting data and it influences how data is reported.

Examples of interval calculation

These examples show how the REDUCE subcommand is used in combination with FROM, TO, INTERVAL, and BOUNDARY to align start times and intervals.

Using REDUCE to align to the start of the hour

```
⋮
REDUCE
  FROM      (,08:00)
  TO        (,10:00)
  INTERVAL  (30)
  BOUNDARY  (60)
⋮
```

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the FROM time (08:00). Subsequent intervals start every 30 minutes (08:30, 09:00, and 09:30 each day).

Using REDUCE to cover a day

```
⋮
REDUCE
  INTERVAL (1440)
```

```

: BOUNDARY (60)
:

```

The following defaults are applied:

- The default for FROM is all dates and a time of 00:00:00.00
- The default for TO is all dates and a time of 23:59:59.99

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the FROM time (00:00). Subsequent intervals cover 1 440 minutes or one day. An interval starts at 00:00 each day.

Using REDUCE to start every hour

```

:
: REDUCE
:   FROM      (,08:30)
:   TO        (,12:00)
:   INTERVAL  (60)
:   BOUNDARY  (60)
: REPORT
:   FROM      (,08:30)
:   TO        (,12:00)
:
:

```

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the hour of the FROM time (08:00). Subsequent intervals start every hour (09:00, 10:00, and 11:00).

Processing intervals

Interval processing within the REDUCE subcommand determines the time intervals that are used for reducing Accounting data and it influences how data is reported.

Use intervals for:

- Reporting by intervals
- Producing reports with different time spans

If SAVE accompanies REDUCE, the reduced data is saved into the Save data set.

This following topics provide examples of interval processing.

“How intervals are calculated” on page 176

The start time of the first interval that is processed by REDUCE is influenced by BOUNDARY, INTERVAL, and FROM.

“Examples of interval processing” on page 176

Sample of SMF File data used for the examples.

“Examples using REDUCE” on page 177

These examples show how the start time is aligned by the use of BOUNDARY.

“Examples using REDUCE and REPORT” on page 178

These examples show the effect of REPORT FROM and TO on REDUCE INTERVAL and BOUNDARY.

“The effect of REDUCE on TRACE” on page 182

These examples show the effect of TRACE FROM and TO on REDUCE BOUNDARY and INTERVAL.

“The effect of REDUCE on FILE” on page 184

These examples show the effect of FILE FROM and TO on REDUCE BOUNDARY and INTERVAL.

How intervals are calculated

The start time of the first interval that is processed by REDUCE is influenced by BOUNDARY, INTERVAL, and FROM.

OMEGAMON XE for DB2 PE attempts to reduce all data that falls between the FROM and TO times. The first interval starts at a time aligned with BOUNDARY, at or before the FROM time. If an interval cannot be aligned with the FROM time, the first properly aligned interval starting *before* the FROM time is used. An interval that starts before the FROM time only contains data between the FROM time and the start of the next interval. Input data before the FROM time is not processed.

Although there is no restriction on the INTERVAL and BOUNDARY combination, your specification should comply with the following rules:

- For intervals of less than 60 (excluding 0), there should be a whole number of intervals in an hour. Choose one of the following values:
 - 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30
- For intervals of 60 or greater, there should be a whole number of intervals in a day. Choose one of the following values:
 - 60, 120, 180, 240, 360, 480, 720, or 1440
- For intervals of one day (1440) or greater, INTERVAL should be a multiple of 1440
- Select your interval and boundary so that the first interval starts at the FROM time

If you do not require interval processing, do not change the default INTERVAL (0) for performance reasons.

Always use the largest interval that meets your reporting requirements. For example, if daily reports provide sufficient granularity, use INTERVAL (1440).

Examples of interval processing

Sample of SMF File data used for the examples.

The following examples assume:

1. The DB2 instrumentation facility is started for Accounting to SMF at DB2 startup by the following DB2 command:
-START TRACE (ACCTG) DEST(SMF) CLASS(1)
2. The system programmer wants to analyze performance for Thursday, 14 March 1999, so the SMF file for that day is obtained.

The following figure represents sample data from the SMF file. Actual trace data can have a different distribution of DB2 records and timestamps. For the purpose of this example, assume that the Statistics trace was previously active.

DB2 Timestamp	Sequence Destination #	FCID	Record Descriptions
7/14/13 8:30:00	0001	0001	System statistics
7/14/13 8:30:01	0002	0002	Database statistics
7/14/13 8:31:05	0003	0003	Accounting
7/14/13 8:45:13	0004	0003	Accounting
7/14/13 8:57:27	0005	0003	Accounting
7/14/13 8:59:59	0006	0001	System statistics
7/14/13 9:00:00	0007	0002	Database statistics
7/14/13 9:12:11	0008	0003	Accounting
7/14/13 9:15:00	0009	0003	Accounting
7/14/13 9:29:59	0010	0001	System statistics
7/14/13 9:30:00	0011	0002	Database statistics
7/14/13 9:30:01	0012	0003	Accounting
7/14/13 9:32:00	0013	0003	Accounting
7/14/13 9:43:00	0014	0003	Accounting
7/14/13 9:55:59	0015	0003	Accounting
7/14/13 10:00:00	0016	0001	System statistics
7/14/13 10:00:00	0017	0002	Database statistics
7/14/13 10:30:00	0018	0001	System statistics
7/14/13 10:30:01	0019	0002	Database statistics
7/14/13 11:00:00	0020	0001	System statistics
7/14/13 11:00:01	0021	0002	Database statistics
7/14/13 11:29:59	0022	0001	System statistics
7/14/13 11:30:00	0023	0002	Database statistics
7/14/13 12:00:01	0024	0001	System statistics
7/14/13 12:00:01	0025	0002	Database statistics
7/14/13 12:15:00	0026	0003	Accounting
7/14/13 12:17:54	0027	0003	Accounting
7/14/13 12:30:00	0028	0001	System statistics
7/14/13 12:30:00	0029	0002	Database statistics
7/14/13 13:00:00	0030	0001	System statistics
7/14/13 13:00:01	0031	0002	Database statistics
7/14/13 13:10:31	0032	0003	Accounting
7/14/13 13:15:00	0033	0003	Accounting
7/14/13 13:17:54	0034	0003	Accounting
7/14/13 13:23:34	0035	0003	Accounting
7/14/13 13:27:00	0036	0003	Accounting
7/14/13 13:30:00	0037	0001	System statistics
7/14/13 13:30:00	0038	0002	Database statistics
7/14/13 13:37:30	0039	0003	Accounting
7/14/13 15:20:00	0040	0001	System statistics
7/14/13 15:20:00	0041	0002	Database statistics
7/14/13 15:37:30	0042	0003	Accounting
7/14/13 15:50:00	0043	0001	System statistics
7/14/13 15:50:01	0044	0002	Database statistics
7/14/13 16:15:00	0045	0001	System statistics
7/14/13 16:15:00	0046	0002	Database statistics
7/14/13 16:28:02	0047	0003	Accounting
7/14/13 16:30:31	0048	0003	Accounting
7/14/13 16:44:00	0049	0003	Accounting
7/14/13 16:44:59	0050	0003	Accounting
7/14/13 16:45:00	0051	0001	System statistics
7/14/13 16:45:01	0052	0002	Database statistics
7/14/13 16:51:06	0053	0003	Accounting
7/14/13 16:54:17	0054	0003	Accounting
7/14/13 16:55:37	0055	0003	Accounting
7/14/13 16:57:59	0056	0003	Accounting

Figure 41. Example SMF File data

3. OMEGAMON XE for DB2 PE is run to get Accounting reports.

Examples using REDUCE

These examples show how the start time is aligned by the use of BOUNDARY.

Example 1

```
⋮  
ACCOUNTING  
  REDUCE  
    FROM      (,09:00)  
    TO        (,11:00)  
    INTERVAL  (30)  
    BOUNDARY  (60)  
⋮
```

The data that is available for reducing is limited by the GLOBAL command. If you specified FROM and TO dates and times in GLOBAL, OMEGAMON XE for DB2 PE discards all records outside those dates and times before reducing. In this example no GLOBAL command is in effect, so the FROM and TO dates and times that are specified with REDUCE are used. No dates are specified, so *all dates* are included.

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the FROM time (09:00). Subsequent intervals start every 30 minutes. The following intervals are calculated:

- Beginning at 9:00:00 - containing two Accounting records
- Beginning at 9:30:00 - containing four Accounting records

No intervals are calculated at 10:00:00 or 10:30:00 because there are no Accounting records.

The interval *begin* and *end* times are reported in the job summary report. The job summary report indicates that an interval is calculated and indicates the number of records processed during the interval. It does not indicate the number of consolidated records after reducing.

Example 2

```
⋮  
ACCOUNTING  
  REDUCE  
    INTERVAL (1440)  
    BOUNDARY (60)  
⋮
```

In this example no GLOBAL command is in effect and no dates or times have been specified in REDUCE, so the following defaults are applied:

- The default for FROM is all dates and a time of 00:00:00.00
- The default for TO is all dates and a time of 23:59:59.99

BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the FROM time (00:00). Subsequent intervals cover 1440 minutes or one day. An interval starts at 00:00 each day. In this example, there is one interval that contains all of the Accounting records in the input data set.

The interval *begin* and *end* times are reported in the job summary report.

Examples using REDUCE and REPORT

These examples show the effect of REPORT FROM and TO on REDUCE INTERVAL and BOUNDARY.

Example 1

```

:
ACCOUNTING
  REDUCE
    INTERVAL (60)
  REPORT
    FROM      (, 9:00:00)
    TO        (,12:00:00)
:
:
```

In this example no GLOBAL command is in effect and no FROM and TO times are specified in the REDUCE subcommand, so all records in the input are reduced.

The REDUCE subcommand specifies that data is reduced at 60-minute intervals. By default, the boundary is set to 60. Data is gathered and consolidated every hour on the hour.

The Accounting data starts at 8:31:05. The INTERVAL begins on the hour. OMEGAMON XE for DB2 PE determines that the record at 8:31:05 falls into an interval beginning at 8:00:00 and sets the beginning interval time to 8:00:00.

Each additional Accounting record is read. The Accounting records at timestamps 8:31:05, 8:45:13, and 8:57:27 all fall into the interval beginning at 8:00:00. These three records are accumulated (added, subtracted, or whatever is appropriate for the given Accounting fields), consolidated, and stored by unique OMEGAMON XE for DB2 PE identifier sets (PRIMAUTH, PLAN, REQLOC, and so on) for the interval beginning at 8:00:00. Messages are written to the job summary report.

OMEGAMON XE for DB2 PE calculates the next interval at 9:00:00. Accounting records at 9:12:11, 9:15:00, 9:30:01, 9:32:00, 9:43:00, and 9:55:59 are accumulated, consolidated and stored for the interval beginning at 9:00:00. Messages are written to the job summary report.

No Accounting records are written during the period from 10:00:00 to 12:00:00, so no intervals are calculated and no messages are written to the job summary report.

Data is stored for the following intervals:

- Beginning at 8:00:00 - containing three Accounting records
- Beginning at 9:00:00 - containing six Accounting records
- Beginning at 12:00:00 - containing two Accounting records
- Beginning at 13:00:00 - containing six Accounting records
- Beginning at 15:00:00 - containing one Accounting record
- Beginning at 16:00:00 - containing eight Accounting records

The systems programmer requested a report with FROM and TO times of 9:00:00 and 12:00:00.

The data that is available for reporting is restricted by the FROM and TO times that are specified in both GLOBAL and REDUCE. If you specify FROM and TO dates and times in GLOBAL, OMEGAMON XE for DB2 PE discards all records outside those dates and times before reducing. If you specify FROM and TO dates and times in REDUCE, all records outside those dates and times are not available to subsequent REPORT subcommands. In this example, no GLOBAL command is in effect and no FROM and TO dates or times are specified in the REDUCE subcommand.

Accounting records that are stored in intervals beginning at or later than 09:00:00, but less than 12:00:00 are accumulated, consolidated, and reported in an Accounting report. In this instance, the report contains Accounting records from 09:12:11 to 09:55:59.

The report header contains the following times:

```
INTERVAL FROM 07/14/13 09:00:00
            TO 07/14/13 10:00:00

REQUESTED ALL DATES 09:00:00
            TO      12:00:00
```

INTERVAL FROM and TO times indicate the actual content of the report. In this instance, the INTERVAL FROM time is 9:00:00, which is the first interval beginning at or later than the REQUESTED FROM time. The INTERVAL TO time is 10:00:00, which is the end time of the last interval beginning at a time less than the REQUESTED TO time.

The REQUESTED FROM and TO times from the REPORT subcommand are printed in the report, even if they are broader than the available data. In this case, ALL DATES is reported in place of an actual date because no date was specified in the REPORT subcommand.

If the requested reporting interval is broader than the available data, the INTERVAL FROM and TO dates and times actually reflect the content of the report. If no FROM and TO dates and times are specified in the REPORT subcommand, the FROM and TO dates and times from GLOBAL are used. If no FROM and TO dates and times are specified in GLOBAL, NOT SPECIFIED is printed.

In “Example 1” on page 179, the INTERVAL FROM and TO times are not the same as the REQUESTED FROM and TO times. However, the data encompassed by the INTERVAL FROM and TO times is the same as the data encompassed by the REQUESTED FROM and TO times. This is not always the case, as in “Example 2.”

Example 2

```
:
:
ACCOUNTING
  REDUCE
    INTERVAL (60)
    BOUNDARY (30)
  REPORT
    FROM      (,9:00:00)
    TO        (,12:00:00)
:
```

This command is identical to the command in the “Example 1” on page 179, except that a BOUNDARY of 30 is included. The results are different.

The Accounting data starts at 8:31:05. The INTERVAL begins on the half hour. OMEGAMON XE for DB2 PE determines that the record at 8:31:05 falls into an interval beginning at 8:30:00 and sets the beginning interval time to 8:30:00. The following intervals are calculated:

- Beginning at 8:30:00 - containing five Accounting records
- Beginning at 9:30:00 - containing four Accounting records
- Beginning at 11:30:00 - containing two Accounting records
- Beginning at 12:30:00 - containing five Accounting records

- Beginning at 13:30:00 - containing one Accounting record
- Beginning at 15:30:00 - containing two Accounting records
- Beginning at 16:30:00 - containing seven Accounting records

An Accounting report is produced. The REQUESTED FROM and TO times are 9:00:00 and 12:00:00. Accounting records that are stored in intervals beginning at or later than 9:00:00, but less than 12:00:00, are included in the report. The intervals beginning at 9:30:00 and 11:30:00 are included in the report. No interval is calculated at 10:30:00 because there are no Accounting records.

In this instance, the report contains Accounting records from 9:30:01 to 12:17:54. Even though the REQUESTED FROM and TO times are the same as in “Example 1” on page 179, the content of the report is different.

The report header contains the following times:

```
INTERVAL FROM 07/14/13 09:30:00
            TO 07/14/13 12:30:00

REQUESTED ALL DATES 09:00:00
            TO        12:00:00
```

In both, “Example 1” on page 179 and “Example 2” on page 180), the INTERVAL FROM and TO dates and times are different from the REQUESTED FROM and TO times. However, this report contains data outside the REQUESTED FROM and TO times.

The INTERVAL FROM time is the *start time* of the first interval beginning at or after the REQUESTED FROM time (9:00:00 in this case), so the first interval that is reported starts at 9:30:00. Consequently, those records that are written between 9:00:00 and 9:30:00 are not included in the report.

The INTERVAL TO time is the *end time* of the last interval beginning before the REQUESTED TO time (12:00:00 in this case), so the last interval that is reported starts at 11:30:00. Consequently, those records that are written between 12:00:00 and 12:30:00 are included in the report. If the interval starts after the REQUESTED FROM time and before the REQUESTED TO time, all data from the interval is included, even if the interval extends beyond the REQUESTED TO time.

Example 3

```
:
:
Accounting
  REDUCE
    FROM    (,09:30)
    TO      (,13:00)
    INTERVAL (60)
    BOUNDARY (60)
  REPORT
    FROM    (,09:30)
    TO      (,13:00)
  SAVE
:
:
```

In this example, no GLOBAL command is in effect, so the FROM and TO dates and times that are specified in REDUCE are used.

REDUCE specifies that data is reduced at 60-minute intervals. BOUNDARY(60) aligns the start time of the intervals at the start of an hour, so the first interval starts at the hour of the FROM time (09:00).

The first interval starts before the FROM time, but only contains data between the FROM time (09:30) and the start time of the next interval (10:00).

The following intervals are calculated:

- Beginning at 09:00:00 - containing four Accounting records
- Beginning at 12:00:00 - containing two Accounting records

Note: No intervals are calculated at 10:00:00 or 11:00:00 because there are no Accounting records.

The interval *begin* and *end* times are reported in the job summary report.

The systems programmer requested a report with FROM and TO times of 9:30:00 and 13:00:00. Accounting records that are stored in intervals beginning at or later than 9:30:00, but less than 13:00:00 are accumulated, consolidated, and reported in an Accounting report. In this example, the report only contains Accounting records from the interval starting at 12:00:00.

All reduced records are written to the Save data set (including the records from the interval starting at 9:00:00 that were excluded from the report).

The report header contains the following times:

```
INTERVAL FROM 07/14/13 12:00:00
              TO 07/14/13 13:00:00

REQUESTED ALL DATES      09:30:00
              TO         13:00:00
```

The INTERVAL FROM and TO times reflect the actual content of the report. The INTERVAL FROM time is the *start time* of the first interval beginning at or after the REQUESTED FROM time (9:30:00 in this case), so the first interval that is reported starts at 12:00:00. The INTERVAL TO time is the *end time* of the last interval beginning before the REQUESTED TO time (13:00:00 in this case).

The REQUESTED FROM and TO times from the REPORT subcommand are printed in the report, even if they are broader than the available data. In this case, ALL DATES is reported in place of an actual date because no date was specified in the REPORT subcommand.

In this example, the INTERVAL FROM and TO times are not the same as the REQUESTED FROM and TO times, although the data encompassed by both sets of times is the same. However, the content of the Save data set is different from the content of the report.

The effect of REDUCE on TRACE

These examples show the effect of TRACE FROM and TO on REDUCE BOUNDARY and INTERVAL.

Example 1

```
:
ACCOUNTING
  REDUCE
    BOUNDARY (30)
    INTERVAL (60)
  REPORT
    FROM      (, 9:00:00)
    TO        (,12:00:00)
  TRACE
```

```

FROM      (,12:00:00)
TO        (,17:00:00)
:

```

The data that is available for tracing, as with reporting, is influenced first by the GLOBAL command. If you specify FROM and TO dates and times in GLOBAL, all records outside those dates and times are discarded before reducing or tracing.

The data that is available for tracing is also influenced by the REDUCE subcommand. If you specify FROM and TO dates and times in REDUCE, all records outside those dates and times are unavailable to subsequent TRACE subcommands. In this example, no GLOBAL command is in effect and no dates and times are specified for REDUCE, so all records in the input data are available to TRACE. The data that is available for tracing is not restricted by INTERVAL or BOUNDARY.

The ACCOUNTING TRACE subcommand specifies data from 12:00:00 to 17:00:00. The Accounting trace header contains the following times:

```

ACTUAL FROM      07/14/13 12:15:00

REQUESTED ALL DATES      12:00:00
              TO          17:00:00

```

No dates are specified, so *all dates* are included. The REQUESTED FROM and TO dates and times from the TRACE subcommand are printed in the trace.

Example 2

```

:
ACCOUNTING
  REDUCE
    FROM (,10:00)
    TO   (15:00)
  REPORT
    FROM (,10:00)
    TO   (15:00)
  TRACE
    FROM (,09:00)
    TO   (17:00)
:

```

In this example, the TRACE FROM time is before the REDUCE FROM time and the TRACE TO time is after the REDUCE TO time. The trace contains records that are written at or after 10:00:00 and before 15:00:00. The Accounting trace header contains the following times:

```

ACTUAL FROM      07/14/13 12:15:00

REQUESTED ALL DATES      09:00:00
              TO          17:00:00

```

No dates are specified, so *all dates* are included. The REQUESTED FROM and TO times from the TRACE subcommand are printed in the trace. However, the available data is limited by the REDUCE FROM and TO times (10:00:00 to 15:00:00). The requested times do not reflect the actual content of the trace.

The ACTUAL FROM time is the timestamp of the first record in the trace. In this example, the first Accounting record after 10:00:00 is at 12:15:00. The trace includes the eight Accounting records between the REDUCE FROM and TO times.

The effect of REDUCE on FILE

These examples show the effect of FILE FROM and TO on REDUCE BOUNDARY and INTERVAL.

Example 1

```
⋮
ACCOUNTING
  REDUCE
    BOUNDARY (30)
    INTERVAL (60)
  REPORT
    FROM      (, 9:00:00)
    TO        (,12:00:00)
  FILE
    FROM      (,12:00:00)
    TO        (,17:00:00)
⋮
```

The data that is available for filing, as with reducing, is influenced first by the GLOBAL command. If you specify FROM and TO dates and times in GLOBAL, all records outside those dates and times are discarded before reducing or filing.

The data that is available for filing is also influenced by the REDUCE subcommand. If you specify FROM and TO dates and times in REDUCE, all records outside those dates and times are unavailable to subsequent FILE subcommands. In this example, no GLOBAL command is in effect and no times are specified for REDUCE, so all records in the input data are available to FILE. The data that is available for filing is not restricted by INTERVAL or BOUNDARY.

The ACCOUNTING FILE subcommand specifies data from 12:00:00 to 15:00:00.

No dates are specified, so *all dates* are included.

Example 2

```
⋮
ACCOUNTING
  REDUCE
    FROM (,10:00)
    TO   (15:00)
  REPORT
    FROM (,10:00)
    TO   (15:00)
  FILE
    FROM (,09:00)
    TO   (17:00)
⋮
```

In “Example 2,” the FILE FROM time is before the REDUCE FROM time and the FILE TO time is after the REDUCE TO time. The file contains records that are written at or after 10:00:00 and before 15:00:00.

No dates are specified, so *all dates* are included. The data that is available for filing is limited by the REDUCE FROM and TO times (10:00:00 to 15:00:00). The requested times do not reflect the actual content of the file.

In “Example 2,” the first Accounting record after 10:00:00 is at 12:15:00. The file includes the eight Accounting records between the REDUCE FROM and TO times.

Member-scope and group-scope reporting

DB2 enables a query to be processed by several members of a data sharing group. Each member can split the work into parallel tasks. Accounting aggregates the parallel activity done in each member and includes it in the originating query activity.

The information in this section only applies to DB2 data sharing environments.

“Member-scope reports”

Member-scope reports present the activity of a data sharing group by member.

“Group-scope reports” on page 187

Group-scope reports show the instrumentation data aggregated by the OMEGAMON XE for DB2 PE identifiers that you specified and by the individual members.

Member-scope reports

Member-scope reports present the activity of a data sharing group by member.

The activity shown for each member consists of the originating query activity and any parallel activity, including any parallel activity that is performed on other members.

The data in member-scope reports is presented by a combination of location, group, subsystem, and member. Whenever one of these values changes, a new page is started and the page number is initialized. The following command produces a member-scope Accounting report shown in “Member-scope Accounting report.” The command uses the SHORT layout and the default order of the OMEGAMON XE for DB2 PE identifiers, namely plan name within primary authorization ID.

```
⋮  
ACCOUNTING  
  REPORT  
⋮
```

Member-scope Accounting report

The following example shows a member-scope Accounting short report resulting from the default order of the OMEGAMON XE for DB2 PE identifiers:

- The data is ordered according to the authorization IDs and plans. All primary authorization IDs and plans from the input data are shown.
- The TOTAL is printed for primary authorization IDs that contain more than one plan.
- GRAND TOTAL is printed at the end of each member if there is more than one first-level identifier reported.

LOCATION: DSN420Y	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 1-1
GROUP: DSN420Y	Accounting REPORT - SHORT	REQUESTED FROM: NOT SPECIFIED
MEMBER: Y42A		TO: NOT SPECIFIED
SUBSYSTEM: Y42A	ORDER: PRIMAUTH-PLANNAME	INTERVAL FROM: 12/03/13 08:05:17.91
DB2 VERSION: V11	SCOPE: MEMBER	TO: 12/03/13 10:29:26.02

PRIMAUTH PLANNAME	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK SUS #LOCKOUT
USRT001	1	0	0.00	0.00	0.00	0.00	1:09.259093	1:09.083514	211.0K	114.00	47.00		
DSNTEP41	0	1	3.00	1.00	1.00	2.00	1:37.745921	1:37.703681	4.00	6508.00			0
USRT001	10	5	0.00	0.00	0.00	0.00	20:01.439219	14:10.483534	188.7K	63646.00	1621.10		
DSNTEP42	0	5	907.2K	0.90	0.40	2.40	20:33.424823	14:30.612146	823.10	6581.00			0
*** TOTAL ***													
USRT001	11	5	0.00	0.00	0.00	0.00	18:18.513753	12:59.447169	190.7K	57870.36	1478.00		
	0	6	824.7K	0.91	0.45	2.36	18:50.181286	13:20.347740	748.64	6574.36			0
USRT002	1	1	0.00	0.00	0.00	0.00	36.295613	36.185947	51541.00	87.00	66.00		
DSNTEP41	0	0	0.00	1.00	0.00	2.00	56.797521	56.780273	11114.00	1790.00			0
USRT002	3	1	0.00	0.00	0.00	0.00	2:27.021948	1:40.436879	24810.67	6653.33	78.00		
DSNTEP42	0	2	200.2K	0.67	0.33	2.00	2:13.387595	1:24.189014	5253.67	627.00			0
*** TOTAL ***													
USRT002	4	2	0.00	0.00	0.00	0.00	1:59.340364	1:24.374146	31493.25	5011.75	75.00		
	0	2	150.1K	0.75	0.25	2.00	1:54.240077	1:17.336829	6718.75	917.75			0
*** GRAND TOTAL ***													
	15	7	0.00	0.00	0.00	0.00	13:57.400849	9:54.094363	148.3K	43774.73	1103.87		
	0	8	644.8K	0.87	0.40	2.27	14:19.263630	10:07.544830	2340.67	5065.93			0

LOCATION: DSN420Y	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 2-1
GROUP: DSN420Y	Accounting REPORT - SHORT	REQUESTED FROM: NOT SPECIFIED
MEMBER: Y42C		TO: NOT SPECIFIED
SUBSYSTEM: Y42C	ORDER: PRIMAUTH-PLANNAME	INTERVAL FROM: 12/03/13 08:07:02.90
DB2 VERSION: V11	SCOPE: MEMBER	TO: 12/03/13 08:08:47.43

PRIMAUTH PLANNAME	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK SUS #LOCKOUT
USRT001	2	0	0.00	0.00	0.00	0.00	50.545437	50.372842	181.2K	10.00	13.00		
DSNTEP42	0	2	2.00	1.00	1.00	2.00	3:25.952714	3:25.915981	0.00	5676.00			0

LOCATION: DSN420Y	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 3-1
GROUP: DSN420Y	Accounting REPORT - SHORT	REQUESTED FROM: NOT SPECIFIED
MEMBER: Y42D		TO: NOT SPECIFIED
SUBSYSTEM: Y42D	ORDER: PRIMAUTH-PLANNAME	INTERVAL FROM: 12/03/13 08:10:34.27
DB2 VERSION: V11	SCOPE: MEMBER	TO: 12/03/13 08:12:20.87

PRIMAUTH PLANNAME	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK SUS #LOCKOUT
USRT001	2	0	0.00	0.00	0.00	0.00	53.137971	52.965162	181.2K	10.00	14.50		
DSNTEP42	0	2	2.00	1.00	1.00	2.00	3:03.516934	3:03.480655	0.00	5676.00			0

LOCATION: DSN420Y	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 4-1
GROUP: DSN420Y	Accounting REPORT - SHORT	REQUESTED FROM: NOT SPECIFIED
MEMBER: Y42E		TO: NOT SPECIFIED
SUBSYSTEM: Y42E	ORDER: PRIMAUTH-PLANNAME	INTERVAL FROM: 12/03/13 08:14:07.67
DB2 VERSION: V11	SCOPE: MEMBER	TO: 12/03/13 08:14:07.67

PRIMAUTH PLANNAME	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK SUS #LOCKOUT
USRT001	1	0	0.00	0.00	0.00	0.00	53.216137	53.051699	181.2K	10.00	13.00		
DSNTEP42	0	1	2.00	1.00	1.00	2.00	4:20.204918	4:20.170015	0.00	5676.00			0

LOCATION: DSN420Y	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 5-1
GROUP: DSN420Y	Accounting REPORT - SHORT	REQUESTED FROM: NOT SPECIFIED
MEMBER: Y42F		TO: NOT SPECIFIED
SUBSYSTEM: Y42F	ORDER: PRIMAUTH-PLANNAME	INTERVAL FROM: 12/03/13 08:15:52.82
DB2 VERSION: V11	SCOPE: MEMBER	TO: 12/03/13 08:17:36.85

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

USRT001	2	0	0.00	0.00	0.00	0.00	50.873688	50.702225	181.2K	10.00	13.00
DSNTEP42	0	2	2.00	1.00	1.00	2.00	3:49.127243	3:49.091691	0.00	5676.00	0

ACCOUNTING REPORT COMPLETE

Group-scope reports

Group-scope reports show the instrumentation data aggregated by the OMEGAMON XE for DB2 PE identifiers that you specified and by the individual members.

The data is presented by a combination of location and group. Whenever either of these values changes, a new page is started and the page number is initialized. The following command produces a group-scope Accounting report shown in "Group-scope Accounting report." The command uses the SHORT layout and the default order of the OMEGAMON XE for DB2 PE identifiers, namely plan names within primary authorization IDs.

```

:
:
ACCOUNTING
  REPORT
    SCOPE(GROUP)
:
:

```

Group-scope Accounting report

Here is an example of a group-scope Accounting report:

- The data is ordered according to the authorization IDs and plans. All primary authorization IDs and plans from the input data are shown. MEMBER is automatically added as the last ORDER identifier.
- The GROUP TOTAL is printed for primary authorization IDs and plans that contain more than one member.
- The TOTAL is printed for primary authorization IDs that contain more than one plan.
- GRAND TOTAL is printed at the end of each group if there is more than one first-level identifier reported.

LOCATION: DSN420Y GROUP: DSN420Y	OMEGAMON XE for DB2 Performance Expert (V5R3M0) Accounting REPORT - SHORT	PAGE: 1-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 12/03/13 08:05:17.91 TO: 12/03/13 10:29:26.02
DB2 VERSION: V11	ORDER: PRIMAUTH-PLANNAME SCOPE: GROUP	

PRIMAUTH PLANNAME MEMBER	#OCCURS #DISTRS	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK #LOCKOUT	SUS
USRT001 DSNTEP41 Y42A	1 0	0 1	0.00 3.00	0.00 1.00	0.00 1.00	0.00 2.00	1:09.259093 1:37.745921	1:09.083514 1:37.703681	211.0K 4.00	114.00 6508.00	47.00 0			
USRT001 DSNTEP42 Y42A	10 0	5 5	0.00 907.2K	0.00 0.90	0.00 0.40	0.00 2.40	20:01.439219 20:33.424823	14:10.483534 14:30.612146	188.7K 823.10	63646.00 6581.00	1621.10 0			
USRT001 DSNTEP42 Y42C	2 0	0 2	0.00 2.00	0.00 1.00	0.00 1.00	0.00 2.00	50.545437 3:25.952714	50.372842 3:25.915981	181.2K 0.00	10.00 5676.00	13.00 0			
USRT001 DSNTEP42 Y42D	2 0	0 2	0.00 2.00	0.00 1.00	0.00 1.00	0.00 2.00	53.137971 3:03.516934	52.965162 3:03.480655	181.2K 0.00	10.00 5676.00	14.50 0			
USRT001 DSNTEP42 Y42E	1 0	0 1	0.00 2.00	0.00 1.00	0.00 1.00	0.00 2.00	53.216137 4:20.204918	53.051699 4:20.170015	181.2K 0.00	10.00 5676.00	13.00 0			
USRT001 DSNTEP42 Y42F	2 0	0 2	0.00 2.00	0.00 1.00	0.00 1.00	0.00 2.00	50.873688 3:49.127243	50.702225 3:49.091691	181.2K 0.00	10.00 5676.00	13.00 0			
*** GROUP TOTAL *** USRT001 DSNTEP42	17 0	5 12	0.00 533.6K	0.00 0.94	0.00 0.65	0.00 2.24	12:08.042501 13:33.626290	8:41.527500 10:00.192243	185.6K 484.18	37442.94 6208.35	959.12 0			
*** TOTAL *** USRT001	18 0	5 13	0.00 504.0K	0.00 0.94	0.00 0.67	0.00 2.22	11:31.443423 12:53.855158	8:16.391723 9:32.276212	187.0K 457.50	35369.11 6225.00	908.44 0			
USRT002 DSNTEP41 Y42A	1 0	1 0	0.00 0.00	0.00 1.00	0.00 0.00	0.00 2.00	36.295613 56.797521	36.185947 56.780273	51541.00 11114.00	87.00 1790.00	66.00 0			
USRT002 DSNTEP42 Y42A	3 0	1 2	0.00 200.2K	0.00 0.67	0.00 0.33	0.00 2.00	2:27.021948 2:13.387595	1:40.436879 1:24.189014	24810.67 5253.67	6653.33 627.00	78.00 0			
*** TOTAL *** USRT002	4 0	2 2	0.00 150.1K	0.00 0.75	0.00 0.25	0.00 2.00	1:59.340364 1:54.240077	1:24.374146 1:17.336829	31493.25 6718.75	5011.75 917.75	75.00 0			

LOCATION: DSN420Y GROUP: DSN420Y	OMEGAMON XE for DB2 Performance Expert (V5R3M0) Accounting REPORT - SHORT							PAGE: 1-2 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 12/03/13 08:05:17.91 TO: 12/03/13 10:29:26.02						
DB2 VERSION: V11	ORDER: PRIMAUTH-PLANNAME SCOPE: GROUP													
PRIMAUTH PLANNAME MEMBER	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK #LOCKOUT	SUS
*** GRAND TOTAL ***	22 0	7 15	0.00 439.6K	0.00 0.91	0.00 0.59	0.00 2.18	9:47.424685 10:53.925143	7:01.479436 8:02.287233	158.7K 1595.91	29849.59 5260.05			756.91 0	

ACCOUNTING REPORT COMPLETE

Chapter 18. General Statistics information

Statistics reports and traces provide you with a way to analyze DB2 Statistics class 1 trace data.

DB2 accumulates statistics for the system services address space, database services address space, and DDF address space. These statistics are accumulated from the time DB2 is started until it is stopped. At a configurable interval, DB2 logs the current statistics values using IFCID 1 and IFCID 2.

Use the Statistics report set to:

- View system-wide statistics for key DB2 components.
- Compare system performance over two or more reporting intervals.
- Assess system-wide performance for individual DB2 subsystems.
- Assess performance data for a group of data sharing DB2 subsystems.
- Summarize system performance data in a single report.

The Statistics report set provides the following functions:

- Traces present the difference (delta) between the Statistics recorded in two consecutive Statistics record pairs.
- Reports summarize Statistics data over one or more user-defined intervals.
- The File data set contains records in a format that are suitable for use by the DB2 load and Spreadsheet Input-Data Generator utilities.
- The Save-File utility changes Save data sets into sequential data sets for use by the DB2 load and Spreadsheet Input-Data Generator utilities.
- The REDUCE subcommand specifies intervals into which Statistics data is accumulated and apportioned. A report can then be produced that is sorted by these intervals.
- The SAVE and RESTORE subcommands are used to consolidate Statistics for a number of Statistics record pairs and then save these record pairs in a Save data set for later restoration and further processing.
- You can control the level of detail of a report by choosing one of the sample layouts or a layout that you have previously tailored.
- You can use exception processing to identify entries with fields that contain values outside thresholds that you have previously specified.
- In data sharing environments you can produce member-specific or group-specific reports.

“Statistics terms” on page 190

This section describes the various terms used in the Statistics report set.

“Input for Statistics reports” on page 193

This section summarizes the DB2 IFCIDs, trace types, and classes for Statistics reports.

“Functions and utilities of the Statistics report set” on page 193

The Statistics report set consists of the REPORT, TRACE, FILE, SAVE, RESTORE, REDUCE functions, and the Save-File and Spreadsheet Input-Data Generator utilities.

“Controlling the level of detail in reports and traces” on page 195

You can specify the level of detail and the layout of Statistics reports and traces by using the LAYOUT subcommand option of the REPORT or TRACE subcommand.

“Member-scope and group-scope reporting” on page 196

DB2 collects the Statistics data on a per-member basis. DB2 Statistics traces have a local scope. To obtain the statistics for all members of a data sharing group, a DB2 Statistics trace has to be started at each of the members. These traces generally have different start and stop times and can have different periods in which the DB2 Statistics records are externalized.

“Exception processing” on page 202

Exception processing identifies Statistics report, trace, and File data set entries with fields that contain values outside the thresholds specified in the Exception Threshold data set. When exception processing is active, the data to be reported is checked against these thresholds.

“Headers used in Statistics” on page 204

The headers in Statistics reports and traces are identical, except that INTERVAL FROM and INTERVAL TO in the report header is replaced by ACTUAL FROM in the trace header and the SCOPE field does not apply to Statistics traces.

“How values are reported” on page 205

OMEGAMON for DB2 PE reports and traces show field values for both delta and interval reporting.

Statistics terms

This section describes the various terms used in the Statistics report set.

“Delta and interval records”

This section defines the differences among various DB2 Statistics counters, such as accumulated values, snapshot values, and high water mark values.

“Delta records” on page 191

The *delta record* is an OMEGAMON XE for DB2 PE term for a set of counters that describes the activity of a DB2 system between two consecutive DB2 Statistics record pairs.

“Interval records” on page 191

Interval record is a term for a set of counters that describes the activity of a DB2 system in a user-specified period of time.

Delta and interval records

This section defines the differences among various DB2 Statistics counters, such as accumulated values, snapshot values, and high water mark values.

When a DB2 Statistics trace is active, DB2 maintains various Statistics counters and externalizes these counters at regular intervals. The main body of Statistics data is shown in IFCIDs 1 and 2. Although there is a small difference in their timestamps, you can assume that these IFCID are externalized simultaneously. These two records are called a *DB2 Statistics record pair*.

Other statistics information is shown in IFCID 199 (buffer pool statistics at data set level), IFCID 225 (DB2 storage statistics), and IFCID 369 (aggregated accounting statistics).

The counters provided in a DB2 Statistics record pair represent the DB2 activity between the time the pair is externalized and the time the DB2 system was last started. A counter is provided in one of the following forms:

- As an *accumulated value* since the DB2 system was last started. For example, the total number of SELECT statements that were executed since the system was last started.

- As a *current* or *snapshot value*. For example, the number of open data sets at the time the DB2 Statistics record pair was externalized.
- As a *maximum* or *high water mark value* the counter has reached since the time the system was last started. For example, the maximum number of open data sets at any time since the system was last started.

The Statistics report set does not report individual DB2 Statistics record pairs (for that purpose use the long Record Trace report). Instead, it calculates deltas between two consecutive DB2 Statistics record pairs and externalizes the delta records in Statistics traces and File data sets. It also uniformly distributes the delta records over user-specified intervals and externalizes the interval records in Statistics reports and Save data sets.

Delta records

The *delta record* is an OMEGAMON XE for DB2 PE term for a set of counters that describes the activity of a DB2 system between two consecutive DB2 Statistics record pairs.

For example, if a DB2 Statistics record pair is externalized at time $t1$ and the next DB2 Statistics record pair is externalized at $t2$, OMEGAMON XE for DB2 PE creates only one delta record, which represents the DB2 system activity between $t1$ and $t2$.

A counter in the delta record, like the counters in the DB2 Statistics record pairs, is provided in one of the following forms:

- Accumulated value. For example, the total number of SELECT statements that are executed between two consecutive DB2 Statistics record pairs. The accumulated value from a delta record is generally smaller than the accumulated value from the DB2 Statistics record pair which marks the end of the delta record and provides the values accumulated since the DB2 system was last started.
- Current or snapshot value. For example, the number of open data sets at the end of the delta record. It is the same as the value provided in the DB2 Statistics record pair which marks the end of the delta record.
- The maximum or high-water mark value the counter reached from the time the system was last started until the end of the delta record, for example, the maximum number of open data sets. It is the same as the value provided in the DB2 Statistics record pair which marks the end of the delta record.

When a delta record is calculated, OMEGAMON for DB2 PE externalizes it to Statistics traces and File data sets depending on your specification.

Interval records

Interval record is a term for a set of counters that describes the activity of a DB2 system in a user-specified period of time.

Unlike the delta record, where the time interval is determined by two consecutive DB2 Statistics record pairs, you can specify the duration of an interval record by using the INTERVAL subcommand option of the REDUCE subcommand.

Interval records do not generally coincide with the delta records. Several delta records can be contained in one interval record. The interval records are not generally aligned with the delta records. The delta records are aligned with DB2

Statistics record pairs, while you can align the interval record by using the BOUNDARY subcommand option of the REDUCE subcommand.

When an interval record is calculated, OMEGAMON XE for DB2 PE externalizes it to Statistics reports and Save data sets depending on your specification.

To understand the relationship between the delta records and interval records, consider the following example taken from a DB2 10 subsystem.

The DB2 Statistics record pairs are generated every minute. IFCID 225 and IFCID 369 are also externalized every minute, while the generation of IFCID 199 is controlled by DB2 system parameter STATIME. If a Statistics trace is started at 9:15 and stopped at 10:15, 61 DB2 Statistics record pairs are generated.

OMEGAMON XE for DB2 PE processes these record pairs and creates 60 one-minute delta records starting every minute from 9:15 to 10:14. You can print these delta records by specifying the TRACE subcommand or store these delta records in a data set that is suitable for loading into DB2 tables by specifying the FILE subcommand.

If you need a report by an hourly basis, you can specify an interval record duration of 60 minutes by using the INTERVAL subcommand option of REDUCE. By default, the interval records are aligned with hour boundaries. If you require a different alignment, use the BOUNDARY subcommand option. Based on this specification, OMEGAMON XE for DB2 PE uniformly distributes the already calculated one-minute delta records into the corresponding interval records. Two 60-minute interval records are created starting at 9:00 and 10:00. The first interval record contains the delta records from 9:15 to 10:00 and the second interval record contains the delta records from 10:00 to 10:15. You can print these interval records by specifying the REPORT subcommand or store these interval records into a data set for later use by specifying the SAVE subcommand.

A counter in the interval record, like the counters in the delta records, is provided in one of the following forms:

- Accumulated value. For example, the total number of SELECT statements that were executed during the period of time specified for the interval record. Generally, this value is an approximation because the interval records are not aligned with the delta records. When an interval record crosses delta record boundaries, and vice versa, the values of the delta record counters are apportioned and uniformly distributed into overlapping interval records.
- The current or snapshot value is an approximation of the counter value at the end of the interval record. It is derived from the delta records' current values and is weighted according to the overlap between the delta records and the interval records.
- The maximum or high-water mark value the counter has reached between the time the system was last started and the end of the last delta record included in the interval record calculation.

To produce a report that shows the Statistics data for each interval record, you first specify the interval-record duration and alignment with the INTERVAL and BOUNDARY subcommand options of the REDUCE subcommand, then the ORDER(INTEGER) subcommand option of the REPORT subcommand.

A special type of the Statistics report is one where all DB2 Statistics record pairs in the input data set are consolidated in one interval record. Such a report is

produced if no INTERVAL subcommand option is specified and the default INTERVAL(0) is assumed. In this case, the BOUNDARY and ORDER(INTEGER) subcommand options do not apply. The start of the interval record is aligned to the first DB2 Statistics record pair, and there is no ordering by intervals because only one interval record is created. For example, if a Statistics trace is started at 9:15 and ended at 11:15, one interval record for that period is created that contains all counters pertinent to that interval, such as the number of INSERT statements executed from 9:15 to 11:15.

Input for Statistics reports

This section summarizes the DB2 IFCIDs, trace types, and classes for Statistics reports.

The following table summarizes the IFCIDs and DB2 trace classes required to produce Statistics reports and traces.

Table 18. IFCIDs used for Statistics reports

Statistics information	DB2 IFCIDs	DB2 trace type and class
System statistics, DB2 statistics	1, 2	Statistics, class 1
Buffer pool data set statistics	199	Statistics, class 8
DB2 storage statistics	225	Statistics, class 6
Aggregated Accounting statistics	369	Statistics, class 9

Use the following DB2 command to collect all Statistics data:

```
-START TRACE (STATISTICS) CLASS (1,6,8,9) DEST (GTF) LOCATION (*)
```

Because the basic unit of processing in the Statistics report set is the delta record, at least two DB2 Statistics record pairs (IFCID 1 and 2) must be present in the input data set before statistics can be presented.

Functions and utilities of the Statistics report set

The Statistics report set consists of the REPORT, TRACE, FILE, SAVE, RESTORE, REDUCE functions, and the Save-File and Spreadsheet Input-Data Generator utilities.

REPORT

REPORT shows interval records that contain DB2 Statistics data aggregated over user-specified periods of time (see “Interval records” on page 191) which generally do not coincide with periods in which DB2 Statistics records are externalized. For example, you can produce reports that show DB2 system activity per hour, per day, or per the entire period in which the DB2 Statistics data is collected.

A special kind of the Statistics reports are exception reports in which only the interval records are presented that contain selected counters that exceed user-defined threshold values (see “Exception processing” on page 202).

The Statistics reports are produced for each DB2 subsystem and DB2 location present in the input data set. In a data sharing environment you can request the Statistics reports on a per-member basis or per-group basis where the Statistics data is aggregated across all members in the data sharing group (see “Member-scope and group-scope reporting” on page 196).

You can also tailor the format of the reports by specifying which report blocks of data and which fields from a report block are included in a report or by defining your own labels and headings associated with the reported fields (see “Controlling the level of detail in reports and traces” on page 195).

TRACE

TRACE shows delta records that contain DB2 Statistics data within periods of time marked by two consecutive DB2 Statistics record pairs (see “Delta records” on page 191).

A special kind of the Statistics traces are exception traces in which only the delta records are presented that contain selected counters that exceed user-defined thresholds (see “Exception processing” on page 202).

The Statistics traces are produced for each DB2 subsystem and DB2 location present in the input data set. In a data sharing environment the Statistics traces are presented for each member of a data sharing group.

Like reports, you can tailor the layout of the traces (see “Controlling the level of detail in reports and traces” on page 195).

FILE FILE stores delta records (the same data structures presented by the TRACE function) into a sequential data set that is suitable for use by the DB2 load utility.

When delta records are in DB2 tables, you can produce tailored reports by using a reporting facility such as Query Management Facility (QMF). The FILE function can also be considered as an alternative way of archiving the DB2 Statistics data in Save data sets. (See “Creating data for the Performance Database and the Performance Warehouse” on page 273.)

Checking for exception conditions is also available in the FILE function, in which case only the delta records are presented that contain selected counters that exceed user-defined thresholds (see “Exception processing” on page 202).

You can also use the File data sets to generate CSV (comma-separated value) input-data. This CSV data can then be transferred to workstations and imported into spreadsheets to improve DB2 performance analysis using graphical representations or pivot tables. (See Chapter 25, “Generating input data for spreadsheets,” on page 275.)

SAVE You can save data sets as follows:

Without CONVERT option

Stores interval records (the same data structures presented by the REPORT function) into a VSAM data set to:

- Archive the Statistics data for producing long-term reports
- Use the Save-File utility to create a sequential data set that is suitable for use by the DB2 load utility.

(See “Creating data for the Performance Database and the Performance Warehouse” on page 273.)

With CONVERT option

Converts and saves reduced data into a sequential data set that can be loaded into DB2 tables.

You can also use the Save-File data sets to generate CSV (comma-separated value) input-data. This CSV data can then be transferred to workstations and imported into spreadsheets to improve DB2 performance analysis

using graphical representations or pivot tables. (See Chapter 25, “Generating input data for spreadsheets,” on page 275.)

RESTORE

RESTORE reloads a previously saved data set to report it with or without new Statistics data. Saved data can be restored and re-saved as often as required.

REDUCE

REDUCE specifies the duration of the interval records that are to be presented in Statistics reports or stored in a Save data set.

You can also control the volume of data to be reported and saved by using the FROM and TO and the INCLUDE and EXCLUDE subcommand options.

Save-File utility

With the Save-File utility you can:

- Migrate Statistics Save data sets from earlier releases.
- Convert Statistics Save data sets into sequential data sets that are suitable for use by the DB2 load utility.

(See “Creating data for the Performance Database and the Performance Warehouse” on page 273.)

Spreadsheet Input Data Generator utility

You can also use the File and Save-File data sets to generate CSV (comma-separated value) input-data. This CSV data can then be transferred to workstations and imported into spreadsheets to improve DB2 performance analysis using graphical representations or pivot tables. (See Chapter 25, “Generating input data for spreadsheets,” on page 275.)

Controlling the level of detail in reports and traces

You can specify the level of detail and the layout of Statistics reports and traces by using the LAYOUT subcommand option of the REPORT or TRACE subcommand.

The following sample layouts are supplied:

- SHORT
- LONG

In Statistics reports and traces, layout SHORT provides general data and layout LONG provides comprehensive data. Layout SHORT contains selected blocks and fields from statistics categories. layout LONG contains most, but not all, of the available blocks and fields.

If the use of commands, subcommands, and subcommand options does not provide sufficient control over the contents of reports and traces, you can use the User-Tailored Reporting (UTR) feature to create and tailor your own report and trace layouts. With UTR you can add, remove, and change individual fields and entire report blocks to control the volume, contents, and layouts of your reports and traces. For more information, see “Tailoring report layouts” on page 236.

Member-scope and group-scope reporting

DB2 collects the Statistics data on a per-member basis. DB2 Statistics traces have a local scope. To obtain the statistics for all members of a data sharing group, a DB2 Statistics trace has to be started at each of the members. These traces generally have different start and stop times and can have different periods in which the DB2 Statistics records are externalized.

The information in this section only applies to DB2 data sharing environments.

“Member-scope reports and traces”

OMEGAMON XE for DB2 PE can process all members of a data sharing group at the same time and produce reports and traces that show the Statistics data on a per-member basis.

“Group-scope reports and traces” on page 198

In group-scope reports, OMEGAMON XE for DB2 PE applies the same user-specified Statistics interval to all members of a group, presents these Statistics intervals for each member, and then adds up the counters across all members and presents the counters as Statistics on a per-group basis.

Member-scope reports and traces

OMEGAMON XE for DB2 PE can process all members of a data sharing group at the same time and produce reports and traces that show the Statistics data on a per-member basis.

Such member-scope reports and traces are like those produced in non-data sharing environments. Reports and traces are produced on a per-subsystem basis.

The following extracts from a long Statistics report demonstrates the concept of member-scope reporting.

- The data sharing group DBN1 consists of two members, SN13 and SN14.
- The statistics are presented separately for each member (note the MEMBER field in the page heading).
- The non-data sharing counters (such as the CPU TIMES block) as well as the data sharing counters (such as the Group Buffer Pools Activity block) are presented. To keep the figure compact only selected Statistics data is shown.
- The default Statistics interval is assumed, which means that the data is presented for the entire period the statistics are available.
- There is no aggregated statistics data for the entire group.

Long Statistics report for member SN13 of group DBN1

The following example shows a partial member-scope long Statistics report for member SN13 of group DBN1.

```
LOCATION: PMODBN1          OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)          PAGE: 2-1
GROUP: DBN1               STATISTICS REPORT - LONG                                REQUESTED FROM: NOT SPECIFIED
MEMBER: SN13                                                       TO: NOT SPECIFIED
SUBSYSTEM: SN13            INTERVAL FROM: 03/27/12 21:53:27.65
DB2 VERSION: V10           TO: 03/28/12 07:53:27.65
                           SCOPE: MEMBER
```

----- HIGHLIGHTS -----

INTERVAL START	: 03/27/12 21:53:27.65	SAMPLING START:	03/27/12 21:53:27.65	TOTAL THREADS	:	187.00
INTERVAL END	: 03/28/12 07:53:27.65	SAMPLING END	: 03/28/12 07:53:27.65	TOTAL COMMITS	:	47.00
INTERVAL ELAPSED:	9:59:59.999970	OUTAGE ELAPSED:	0.000000	DATA SHARING MEMBER:	:	N/A

SQL DML	QUANTITY	/SECOND	/THREAD	/COMMIT	SQL DCL	QUANTITY	/SECOND	/THREAD	/COMMIT
SELECT	56.00	0.00	0.30	1.19	LOCK TABLE	0.00	0.00	0.00	0.00
INSERT	0.00	0.00	0.00	0.00	GRANT	0.00	0.00	0.00	0.00
NUMBER OF ROWS	0.00	0.00	0.00	0.00	REVOKE	0.00	0.00	0.00	0.00
UPDATE	0.00	0.00	0.00	0.00	SET HOST VARIABLE	2.00	0.00	0.01	0.04

NUMBER OF ROWS	0.00	0.00	0.00	0.00	SET CURRENT SQLID	0.00	0.00	0.00	0.00
MERGE	0.00	0.00	0.00	0.00	SET CURRENT DEGREE	0.00	0.00	0.00	0.00
DELETE	0.00	0.00	0.00	0.00	SET CURRENT RULES	0.00	0.00	0.00	0.00
NUMBER OF ROWS	0.00	0.00	0.00	0.00	SET CURRENT PATH	0.00	0.00	0.00	0.00
					SET CURRENT PRECISION	0.00	0.00	0.00	0.00
PREPARE	42.00	0.00	0.22	0.89					
DESCRIBE	25.00	0.00	0.13	0.53	CONNECT TYPE 1	8.00	0.00	0.04	0.17
DESCRIBE TABLE	0.00	0.00	0.00	0.00	CONNECT TYPE 2	1.00	0.00	0.01	0.02
OPEN	18.00	0.00	0.10	0.38	RELEASE	0.00	0.00	0.00	0.00
CLOSE	18.00	0.00	0.10	0.38	SET CONNECTION	0.00	0.00	0.00	0.00
FETCH	26363.00	0.73	140.98	560.91					
NUMBER OF ROWS	26358.00	0.73	140.95	560.81	ASSOCIATE LOCATORS	0.00	0.00	0.00	0.00
					ALLOCATE CURSOR	0.00	0.00	0.00	0.00
TOTAL DML	26522.00	0.74	141.83	564.30					
					HOLD LOCATOR	0.00	0.00	0.00	0.00
					FREE LOCATOR	0.00	0.00	0.00	0.00
					TOTAL	11.00	0.00	0.06	0.23
STORED PROCEDURES	QUANTITY	/SECOND	/THREAD	/COMMIT	TRIGGERS	QUANTITY	/SECOND	/THREAD	/COMMIT
CALL STATEMENT EXECUTED	0.00	0.00	0.00	0.00	STATEMENT TRIGGER ACTIVATED	0.00	0.00	0.00	0.00
PROCEDURE ABENDED	0.00	0.00	0.00	0.00	ROW TRIGGER ACTIVATED	0.00	0.00	0.00	0.00
CALL STATEMENT TIMED OUT	0.00	0.00	0.00	0.00	SQL ERROR OCCURRED	0.00	0.00	0.00	0.00
CALL STATEMENT REJECTED	0.00	0.00	0.00	0.00					
USER DEFINED FUNCTIONS	QUANTITY	/SECOND	/THREAD	/COMMIT	ROW ID	QUANTITY	/SECOND	/THREAD	/COMMIT
EXECUTED	0.00	0.00	0.00	0.00	DIRECT ACCESS	0.00	0.00	0.00	0.00
ABENDED	0.00	0.00	0.00	0.00	INDEX USED	0.00	0.00	0.00	0.00
TIMED OUT	0.00	0.00	0.00	0.00	TABLE SPACE SCAN USED	0.00	0.00	0.00	0.00
REJECTED	0.00	0.00	0.00	0.00					
USE CURRENTLY COMMITTED	QUANTITY	/SECOND	/THREAD	/COMMIT					
INSERT ROWS SKIPPED	N/A	N/A	N/A	N/A					
DELETE ROWS ACCESSED	N/A	N/A	N/A	N/A					
UPDATE ROWS ACCESSED	N/A	N/A	N/A	N/A					

Long Statistics report for member SN14 of group DBN1

The following example shows a partial member-scope long Statistics report for member SN14 of group DBN1.

LOCATION: PMOBDN1	OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)				PAGE: 3-1				
GROUP: DBN1	STATISTICS REPORT - LONG				REQUESTED FROM: NOT SPECIFIED				
MEMBER: SN14					TO: NOT SPECIFIED				
SUBSYSTEM: SN14					INTERVAL FROM: 03/27/12 22:06:15.94				
DB2 VERSION: V10	SCOPE: MEMBER				TO: 03/28/12 07:36:15.94				
----- HIGHLIGHTS -----									
INTERVAL START : 03/27/12 22:06:15.94		SAMPLING START: 03/27/12 22:06:15.94		TOTAL THREADS		:		293.00	
INTERVAL END : 03/28/12 07:36:15.94		SAMPLING END : 03/28/12 07:36:15.94		TOTAL COMMITS		:		2434.00	
INTERVAL ELAPSED: 9:30:00.000098		OUTAGE ELAPSED: 0.000000		DATA SHARING MEMBER:				N/A	
SQL DML	QUANTITY	/SECOND	/THREAD	/COMMIT	SQL DCL	QUANTITY	/SECOND	/THREAD	/COMMIT
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
SELECT	14.00	0.00	0.05	0.01	LOCK TABLE	0.00	0.00	0.00	0.00
INSERT	0.00	0.00	0.00	0.00	GRANT	0.00	0.00	0.00	0.00
NUMBER OF ROWS	0.00	0.00	0.00	0.00	REVOKE	0.00	0.00	0.00	0.00
UPDATE	1422.00	0.04	4.85	0.58	SET HOST VARIABLE	4.00	0.00	0.01	0.00
NUMBER OF ROWS	0.00	0.00	0.00	0.00	SET CURRENT SQLID	2.00	0.00	0.01	0.00
MERGE	0.00	0.00	0.00	0.00	SET CURRENT DEGREE	0.00	0.00	0.00	0.00
DELETE	0.00	0.00	0.00	0.00	SET CURRENT RULES	0.00	0.00	0.00	0.00
NUMBER OF ROWS	0.00	0.00	0.00	0.00	SET CURRENT PATH	0.00	0.00	0.00	0.00
					SET CURRENT PRECISION	0.00	0.00	0.00	0.00
PREPARE	2266.00	0.07	7.73	0.93					
DESCRIBE	7.00	0.00	0.02	0.00	CONNECT TYPE 1	0.00	0.00	0.00	0.00
DESCRIBE TABLE	0.00	0.00	0.00	0.00	CONNECT TYPE 2	0.00	0.00	0.00	0.00
OPEN	1435.00	0.04	4.90	0.59	RELEASE	0.00	0.00	0.00	0.00
CLOSE	13.00	0.00	0.04	0.01	SET CONNECTION	0.00	0.00	0.00	0.00
FETCH	14604.00	0.43	49.84	6.00					
NUMBER OF ROWS	13175.00	0.39	44.97	5.41	ASSOCIATE LOCATORS	0.00	0.00	0.00	0.00
					ALLOCATE CURSOR	0.00	0.00	0.00	0.00
TOTAL DML	19761.00	0.58	67.44	8.12					
					HOLD LOCATOR	0.00	0.00	0.00	0.00
					FREE LOCATOR	0.00	0.00	0.00	0.00
					TOTAL	6.00	0.00	0.02	0.00
STORED PROCEDURES	QUANTITY	/SECOND	/THREAD	/COMMIT	TRIGGERS	QUANTITY	/SECOND	/THREAD	/COMMIT
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
CALL STATEMENT EXECUTED	0.00	0.00	0.00	0.00	STATEMENT TRIGGER ACTIVATED	0.00	0.00	0.00	0.00
PROCEDURE ABENDED	0.00	0.00	0.00	0.00	ROW TRIGGER ACTIVATED	0.00	0.00	0.00	0.00
CALL STATEMENT TIMED OUT	0.00	0.00	0.00	0.00	SQL ERROR OCCURRED	0.00	0.00	0.00	0.00
CALL STATEMENT REJECTED	0.00	0.00	0.00	0.00					

USER DEFINED FUNCTIONS	QUANTITY	/SECOND	/THREAD	/COMMIT	ROW ID	QUANTITY	/SECOND	/THREAD	/COMMIT
EXECUTED	0.00	0.00	0.00	0.00	DIRECT ACCESS	0.00	0.00	0.00	0.00
ABENDED	0.00	0.00	0.00	0.00	INDEX USED	0.00	0.00	0.00	0.00
TIMED OUT	0.00	0.00	0.00	0.00	TABLE SPACE SCAN USED	0.00	0.00	0.00	0.00
REJECTED	0.00	0.00	0.00	0.00					

USE CURRENTLY COMMITTED	QUANTITY	/SECOND	/THREAD	/COMMIT
INSERT ROWS SKIPPED	N/A	N/A	N/A	N/A
DELETE ROWS ACCESSED	N/A	N/A	N/A	N/A
UPDATE ROWS ACCESSED	N/A	N/A	N/A	N/A

Group-scope reports and traces

In group-scope reports, OMEGAMON XE for DB2 PE applies the same user-specified Statistics interval to all members of a group, presents these Statistics intervals for each member, and then adds up the counters across all members and presents the counters as Statistics on a per-group basis.

Most Statistics counters are pertinent to the member that maintains them, for example, the counters that describe the local buffer pool activity. There are, however, some Statistics counters that are also pertinent to the entire data sharing group, for example, the counters that describes the global locking activity and group-buffer-pools usage. To present these counters on a per-group basis, OMEGAMON XE for DB2 PE combines the Statistics data of the individual members and presents it for the entire group. There are no group-scope traces because DB2 Statistics traces are not synchronized across a data sharing group. They have generally different start and stop times and frequency of externalizing the Statistics data.

The following extract from a short Statistics report demonstrates the concept of group-scope reporting.

- The statistics are presented separately for each member (note the MEMBER field in the HIGHLIGHTS block).
- Only selected counters are shown (HIGHLIGHTS, DATA SHARING LOCKS, and GROUP BP0), to keep the figure compact.
- After the members' Statistics are presented, the aggregated Statistics data for the entire group is shown.
 - The MEMBER field in the HIGHLIGHTS block displays the number of members for which the Statistics are aggregated.
 - The group's Statistics counters are calculated by adding up the members' Statistics counters.
 - As the default Statistics interval is assumed (the data is presented for the entire period the statistics are available), the statistics are provided from the earliest to the latest time for which the statistics data is available.
 - The group's interval elapsed time is the average elapsed time of the members' interval elapsed times.

Group-scope short Statistics report for member SN13 of group DBN1

The following example shows a group-scope short Statistics report for member SN13 of group DBN1.

LOCATION: PMODBN1
GROUP: DBN1

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
STATISTICS REPORT - SHORT

PAGE: 2-1
REQUESTED FROM: NOT SPECIFIED
TO: NOT SPECIFIED
INTERVAL FROM: 03/27/12 21:53:27.65
TO: 03/28/12 07:53:27.65

DB2 VERSION: V10

SCOPE: GROUP

```

----- HIGHLIGHTS -----
INTERVAL START: 03/27/12 21:53:27.65   INTERVAL ELAPSED: 10:00:00.00   INCREMENTAL BINDS      :      0.00   DBAT QUEUED:      N/P
INTERVAL END   : 03/28/12 07:53:27.65   OUTAGE ELAPSED  :      0.000000   AUTH SUCC.W/OUT CATALOG:    178.00   DB2 COMMAND:    185.00
SAMPLING START: 03/27/12 21:53:27.65   TOTAL THREADS   :      187.00   BUFF.UPDT/PAGES WRITTEN:      N/C   TOTAL API   :    5365.00
SAMPLING END   : 03/28/12 07:53:27.65   TOTAL COMMITS   :       47.00   PAGES WRITTEN/WRITE I/O:      N/C   MEMBER      :    SN13

```

```

DATA SHARING LOCKS  QUANTITY
-----
GLB CONT.RATE (%)    1.05
FLS CONT.RATE (%)    0.00
P/L-LOCKS XES (%)    85.64
LOCK REQ.(P-LOCK)    604.00
UNLOCK REQ.(P-LCK)   305.00
CHANGE REQ.(P-LCK)   300.00
SYNC.XES - LOCK      38938.00
SYNC.XES - CHANGE     84.00
SYNC.XES - UNLOCK    39501.00
BACK.XES-CHILD LCK    20.00
TOTAL SUSPENDS       839.00
P-LCK/NFY ENG.UNAV    0.00
INCOM.RETAINED LCK    0.00
PSET/PART NEGOTIAT   150.00
PAGE NEGOTIATION      1.00

```

LOCATION: PMODBN1
GROUP: DBN1

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
STATISTICS REPORT - SHORT

PAGE: 2-2
REQUESTED FROM: NOT SPECIFIED
TO: NOT SPECIFIED
INTERVAL FROM: 03/27/12 21:53:27.65
TO: 03/28/12 07:53:27.65

DB2 VERSION: V10

SCOPE: GROUP

```

----- HIGHLIGHTS -----
INTERVAL START: 03/27/12 21:53:27.65   INTERVAL ELAPSED: 10:00:00.00   INCREMENTAL BINDS      :      0.00   DBAT QUEUED:      N/P
INTERVAL END   : 03/28/12 07:53:27.65   OUTAGE ELAPSED  :      0.000000   AUTH SUCC.W/OUT CATALOG:    178.00   DB2 COMMAND:    185.00
SAMPLING START: 03/27/12 21:53:27.65   TOTAL THREADS   :      187.00   BUFF.UPDT/PAGES WRITTEN:      N/C   TOTAL API   :    5365.00
SAMPLING END   : 03/28/12 07:53:27.65   TOTAL COMMITS   :       47.00   PAGES WRITTEN/WRITE I/O:      N/C   MEMBER      :    SN13

```

GROUP BP0	QUANTITY	GROUP BP8K	QUANTITY	GROUP BP16K	QUANTITY
GROUP BP R/W RATIO (%)	23.66	GROUP BP R/W RATIO (%)	N/C	GROUP BP R/W RATIO (%)	N/C
GBP-DEPENDENT GETPAGES	462.00	GBP-DEPENDENT GETPAGES	85.00	GBP-DEPENDENT GETPAGES	0.00
SYN.READ(XI)-DATA RETURNED	51.00	SYN.READ(XI)-DATA RETURNED	2.00	SYN.READ(XI)-DATA RETURNED	0.00
SYN.READ(XI)-NO DATA RETURN	272.00	SYN.READ(XI)-NO DATA RETURN	12.00	SYN.READ(XI)-NO DATA RETURN	0.00
SYN.READ(NF)-DATA RETURNED	2.00	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
CLEAN PAGES ASYN.WRT	0.00	CLEAN PAGES ASYN.WRT	0.00	CLEAN PAGES ASYN.WRT	0.00
CHANGED PGS SYN.WRTN	224.00	CHANGED PGS SYN.WRTN	0.00	CHANGED PGS SYN.WRTN	0.00
CHANGED PGS ASYN.WRT	0.00	CHANGED PGS ASYN.WRT	0.00	CHANGED PGS ASYN.WRT	0.00
REG.PG LIST (RPL) RQ	0.00	REG.PG LIST (RPL) RQ	0.00	REG.PG LIST (RPL) RQ	0.00
CHANGED PGS READ RPL	0.00	CHANGED PGS READ RPL	0.00	CHANGED PGS READ RPL	0.00
PAGES CASTOUT	180.00	PAGES CASTOUT	0.00	PAGES CASTOUT	0.00
WRITE AND REGISTER	180.00	WRITE AND REGISTER	0.00	WRITE AND REGISTER	0.00
WRITE AND REGISTER MULT	20.00	WRITE AND REGISTER MULT	0.00	WRITE AND REGISTER MULT	0.00
READ FOR CASTOUT MULT	15.00	READ FOR CASTOUT MULT	0.00	READ FOR CASTOUT MULT	0.00
PAGES WRITE & REG MULT	44.00	PAGES WRITE & REG MULT	0.00	PAGES WRITE & REG MULT	0.00
EXPLICIT X-INVALIDATIONS	0.00	EXPLICIT X-INVALIDATIONS	0.00	EXPLICIT X-INVALIDATIONS	0.00
CASTOUT CLASS THRESH	0.00	CASTOUT CLASS THRESH	0.00	CASTOUT CLASS THRESH	0.00
GROUP BP CAST.THRESH	0.00	GROUP BP CAST.THRESH	0.00	GROUP BP CAST.THRESH	0.00
WRITE FAILED-NO STOR	0.00	WRITE FAILED-NO STOR	0.00	WRITE FAILED-NO STOR	0.00

```

GROUP TOTAL  QUANTITY
-----
GROUP BP R/W RATIO (%)    24.55
GBP-DEPENDENT GETPAGES    547.00
SYN.READ(XI)-DATA RETURNED  53.00
SYN.READ(XI)-NO DATA RETURN 284.00
SYN.READ(NF)-DATA RETURNED  2.00
SYN.READ(NF)-NO DATA RETURN 0.00
CLEAN PAGES SYN.WRTN       0.00
CLEAN PAGES ASYN.WRT       0.00
CHANGED PGS SYN.WRTN       224.00
CHANGED PGS ASYN.WRT       0.00
REG.PG LIST (RPL) RQ       0.00
CHANGED PGS READ RPL       0.00
PAGES CASTOUT              180.00
WRITE AND REGISTER          180.00
WRITE AND REGISTER MULT     20.00
READ FOR CASTOUT MULT       15.00
PAGES WRITE & REG MULT       44.00
EXPLICIT X-INVALIDATIONS    0.00
CASTOUT CLASS THRESH        0.00
GROUP BP CAST.THRESH        0.00
WRITE FAILED-NO STOR        0.00

```

Group-scope short Statistics report for member SN14 of group DBN1

The following example shows a group-scope short Statistics report for member SN14 of group DBN1.

LOCATION: PMODBNI
GROUP: DBN1

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
STATISTICS REPORT - SHORT

PAGE: 2-3
REQUESTED FROM: NOT SPECIFIED
TO: NOT SPECIFIED
INTERVAL FROM: 03/27/12 21:53:27.65
TO: 03/28/12 07:53:27.65

DB2 VERSION: V10

SCOPE: GROUP

```
----- HIGHLIGHTS -----
INTERVAL START: 03/27/12 22:06:15.94   INTERVAL ELAPSED: 9:30:00.000   INCREMENTAL BINDS      :      0.00   DBAT QUEUED:      N/P
INTERVAL END   : 03/28/12 07:36:15.94   OUTAGE ELAPSED      :      0.000000   AUTH SUCC.W/OUT CATALOG: 277.00   DB2 COMMAND: 357.00
SAMPLING START: 03/27/12 22:06:15.94   TOTAL THREADS      :      293.00   BUFF.UPDT/PAGES WRITTEN: 9.83   TOTAL API   : 16016.00
SAMPLING END   : 03/28/12 07:36:15.94   TOTAL COMMITS      :      984.00   PAGES WRITTEN/WRITE I/O: 1.00   MEMBER      : SN14
```

```
DATA SHARING LOCKS  QUANTITY
-----
GLB CONT.RATE (%)    1.27
FLS CONT.RATE (%)    0.03
P/L-LOCKS XES (%)    69.70
LOCK REQ.(P-LOCK)    616.00
UNLOCK REQ.(P-LCK)   354.00
CHANGE REQ.(P-LCK)   255.00
SYNC.XES - LOCK      38362.00
SYNC.XES - CHANGE    1020.00
SYNC.XES - UNLOCK    39447.00
BACK.XES-CHILD LCK   42.00
TOTAL SUSPENDS       1019.00
P-LCK/NFY ENG.UNAV    0.00
INCOM.RETAINED LCK   0.00
PSET/PART NEGOTIAT    114.00
PAGE NEGOTIATION      0.00
```

LOCATION: PMODBNI
GROUP: DBN1

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
STATISTICS REPORT - SHORT

PAGE: 2-4
REQUESTED FROM: NOT SPECIFIED
TO: NOT SPECIFIED
INTERVAL FROM: 03/27/12 21:53:27.65
TO: 03/28/12 07:53:27.65

DB2 VERSION: V10

SCOPE: GROUP

```
----- HIGHLIGHTS -----
INTERVAL START: 03/27/12 22:06:15.94   INTERVAL ELAPSED: 9:30:00.000   INCREMENTAL BINDS      :      0.00   DBAT QUEUED:      N/P
INTERVAL END   : 03/28/12 07:36:15.94   OUTAGE ELAPSED      :      0.000000   AUTH SUCC.W/OUT CATALOG: 277.00   DB2 COMMAND: 357.00
SAMPLING START: 03/27/12 22:06:15.94   TOTAL THREADS      :      293.00   BUFF.UPDT/PAGES WRITTEN: 9.83   TOTAL API   : 16016.00
SAMPLING END   : 03/28/12 07:36:15.94   TOTAL COMMITS      :      984.00   PAGES WRITTEN/WRITE I/O: 1.00   MEMBER      : SN14
```

GROUP BP0	QUANTITY	GROUP BP8K	QUANTITY	GROUP BP16K	QUANTITY
GROUP BP R/W RATIO (%)	19.21	GROUP BP R/W RATIO (%)	0.00	GROUP BP R/W RATIO (%)	N/C
GBP-DEPENDENT GETPAGES	743.00	GBP-DEPENDENT GETPAGES	73.00	GBP-DEPENDENT GETPAGES	0.00
SYN.READ(XI)-DATA RETURNED	38.00	SYN.READ(XI)-DATA RETURNED	0.00	SYN.READ(XI)-DATA RETURNED	0.00
SYN.READ(XI)-NO DATA RETURN	160.00	SYN.READ(XI)-NO DATA RETURN	1.00	SYN.READ(XI)-NO DATA RETURN	0.00
SYN.READ(NF)-DATA RETURNED	1.00	SYN.READ(NF)-DATA RETURNED	0.00	SYN.READ(NF)-DATA RETURNED	0.00
SYN.READ(NF)-NO DATA RETURN	20.00	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
CLEAN PAGES ASYN.WRT	0.00	CLEAN PAGES ASYN.WRT	0.00	CLEAN PAGES ASYN.WRT	0.00
CHANGED PGS SYN.WRTN	203.00	CHANGED PGS SYN.WRTN	13.00	CHANGED PGS SYN.WRTN	0.00
CHANGED PGS ASYN.WRT	0.00	CHANGED PGS ASYN.WRT	0.00	CHANGED PGS ASYN.WRT	0.00
REG.PG LIST (RPL) RQ	3.00	REG.PG LIST (RPL) RQ	0.00	REG.PG LIST (RPL) RQ	0.00
CHANGED PGS READ RPL	0.00	CHANGED PGS READ RPL	0.00	CHANGED PGS READ RPL	0.00
PAGES CASTOUT	144.00	PAGES CASTOUT	13.00	PAGES CASTOUT	0.00
WRITE AND REGISTER	128.00	WRITE AND REGISTER	0.00	WRITE AND REGISTER	0.00
WRITE AND REGISTER MULT	25.00	WRITE AND REGISTER MULT	3.00	WRITE AND REGISTER MULT	0.00
READ FOR CASTOUT MULT	16.00	READ FOR CASTOUT MULT	3.00	READ FOR CASTOUT MULT	0.00
PAGES WRITE & REG MULT	75.00	PAGES WRITE & REG MULT	13.00	PAGES WRITE & REG MULT	0.00
EXPLICIT X-INVALIDATIONS	0.00	EXPLICIT X-INVALIDATIONS	0.00	EXPLICIT X-INVALIDATIONS	0.00
CASTOUT CLASS THRESH	0.00	CASTOUT CLASS THRESH	1.00	CASTOUT CLASS THRESH	0.00
GROUP BP CAST.THRESH	0.00	GROUP BP CAST.THRESH	0.00	GROUP BP CAST.THRESH	0.00
WRITE FAILED-NO STOR	0.00	WRITE FAILED-NO STOR	0.00	WRITE FAILED-NO STOR	0.00

```
GROUP TOTAL  QUANTITY
-----
GROUP BP R/W RATIO (%)    18.06
GBP-DEPENDENT GETPAGES    816.00
SYN.READ(XI)-DATA RETURNED 38.00
SYN.READ(XI)-NO DATA RETURN 161.00
SYN.READ(NF)-DATA RETURNED 1.00
SYN.READ(NF)-NO DATA RETURN 20.00
CLEAN PAGES SYN.WRTN      0.00
CLEAN PAGES ASYN.WRT      0.00
CHANGED PGS SYN.WRTN      216.00
CHANGED PGS ASYN.WRT      0.00
REG.PG LIST (RPL) RQ      3.00
CHANGED PGS READ RPL      0.00
PAGES CASTOUT             157.00
WRITE AND REGISTER        128.00
```

WRITE AND REGISTER MULT	28.00
READ FOR CASTOUT MULT	19.00
PAGES WRITE & REG MULT	88.00
EXPLICIT X-INVALIDATIONS	0.00
CASTOUT CLASS THRESH	1.00
GROUP BP CAST.THRESH	0.00
WRITE FAILED-NO STOR	0.00

Group-scope short Statistics report for group DBN1

The following example shows a group-scope short Statistics report for group DBN1.

LOCATION: PMOBDN1	OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)	PAGE: 2-5
GROUP: DBN1	STATISTICS REPORT - SHORT	REQUESTED FROM: NOT SPECIFIED
		TO: NOT SPECIFIED
DB2 VERSION: V10	SCOPE: GROUP	INTERVAL FROM: 03/27/12 21:53:27.65
		TO: 03/28/12 07:53:27.65

---- HIGHLIGHTS -----			
INTERVAL START: 03/27/12 21:53:27.65	INTERVAL ELAPSED: 9:45:00.000	INCREMENTAL BINDS : 0.00	DBAT QUEUED: N/P
INTERVAL END : 03/28/12 07:53:27.65	OUTAGE ELAPSED : 0.000000	AUTH SUCC.W/OUT CATALOG: 455.00	DB2 COMMAND: 542.00
SAMPLING START: 03/27/12 21:53:27.65	TOTAL THREADS : 480.00	BUFF.UPDT/PAGES WRITTEN: 17.78	TOTAL API : 21381.00
SAMPLING END : 03/28/12 07:53:27.65	TOTAL COMMITS : 1031.00	PAGES WRITTEN/WRITE I/O: 1.00	MEMBER : 2

DATA SHARING LOCKS	QUANTITY
GLB CONT.RATE (%)	1.16
FLS CONT.RATE (%)	0.01
P/L-LOCKS XES (%)	76.91
LOCK REQ.(P-LOCK)	1220.00
UNLOCK REQ.(P-LCK)	659.00
CHANGE REQ.(P-LCK)	555.00
SYNC.XES - LOCK	77300.00
SYNC.XES - CHANGE	1104.00
SYNC.XES - UNLOCK	78948.00
BACK.XES-CHILD LCK	62.00
TOTAL SUSPENDS	1858.00
P-LCK/NFY ENG.UNAV	0.00
INCOM.RETAINED LCK	0.00
PSET/PART NEGOTIAT	264.00
PAGE NEGOTIATION	1.00

LOCATION: PMOBDN1	OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)	PAGE: 2-6
GROUP: DBN1	STATISTICS REPORT - SHORT	REQUESTED FROM: NOT SPECIFIED
		TO: NOT SPECIFIED
DB2 VERSION: V10	SCOPE: GROUP	INTERVAL FROM: 03/27/12 21:53:27.65
		TO: 03/28/12 07:53:27.65

---- HIGHLIGHTS -----			
INTERVAL START: 03/27/12 21:53:27.65	INTERVAL ELAPSED: 9:45:00.000	INCREMENTAL BINDS : 0.00	DBAT QUEUED: N/P
INTERVAL END : 03/28/12 07:53:27.65	OUTAGE ELAPSED : 0.000000	AUTH SUCC.W/OUT CATALOG: 455.00	DB2 COMMAND: 542.00
SAMPLING START: 03/27/12 21:53:27.65	TOTAL THREADS : 480.00	BUFF.UPDT/PAGES WRITTEN: 17.78	TOTAL API : 21381.00
SAMPLING END : 03/28/12 07:53:27.65	TOTAL COMMITS : 1031.00	PAGES WRITTEN/WRITE I/O: 1.00	MEMBER : 2

GROUP BP0	QUANTITY	GROUP BP8K	QUANTITY	GROUP BP16K	QUANTITY
GROUP BP R/W RATIO (%)	21.55	GROUP BP R/W RATIO (%)	15.38	GROUP BP R/W RATIO (%)	N/C
GBP-DEPENDENT GETPAGES	1205.00	GBP-DEPENDENT GETPAGES	158.00	GBP-DEPENDENT GETPAGES	0.00
SYN.READ(XI)-DATA RETURNED	89.00	SYN.READ(XI)-DATA RETURNED	2.00	SYN.READ(XI)-DATA RETURNED	0.00
SYN.READ(XI)-NO DATA RETURN	432.00	SYN.READ(XI)-NO DATA RETURN	13.00	SYN.READ(XI)-NO DATA RETURN	0.00
SYN.READ(NF)-DATA RETURNED	3.00	SYN.READ(NF)-DATA RETURNED	0.00	SYN.READ(NF)-DATA RETURNED	0.00
SYN.READ(NF)-NO DATA RETURN	20.00	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
CLEAN PAGES ASYN.WRT	0.00	CLEAN PAGES ASYN.WRT	0.00	CLEAN PAGES ASYN.WRT	0.00
CHANGED PGS SYN.WRTN	427.00	CHANGED PGS SYN.WRTN	13.00	CHANGED PGS SYN.WRTN	0.00
CHANGED PGS ASYN.WRT	0.00	CHANGED PGS ASYN.WRT	0.00	CHANGED PGS ASYN.WRT	0.00
REG.PG LIST (RPL) RQ	3.00	REG.PG LIST (RPL) RQ	0.00	REG.PG LIST (RPL) RQ	0.00
CHANGED PGS READ RPL	0.00	CHANGED PGS READ RPL	0.00	CHANGED PGS READ RPL	0.00
PAGES CASTOUT	324.00	PAGES CASTOUT	13.00	PAGES CASTOUT	0.00
WRITE AND REGISTER	308.00	WRITE AND REGISTER	0.00	WRITE AND REGISTER	0.00
WRITE AND REGISTER MULT	45.00	WRITE AND REGISTER MULT	3.00	WRITE AND REGISTER MULT	0.00
READ FOR CASTOUT MULT	31.00	READ FOR CASTOUT MULT	3.00	READ FOR CASTOUT MULT	0.00
PAGES WRITE & REG MULT	119.00	PAGES WRITE & REG MULT	13.00	PAGES WRITE & REG MULT	0.00
EXPLICIT X-INVALIDATIONS	0.00	EXPLICIT X-INVALIDATIONS	0.00	EXPLICIT X-INVALIDATIONS	0.00
CASTOUT CLASS THRESH	0.00	CASTOUT CLASS THRESH	1.00	CASTOUT CLASS THRESH	0.00
GROUP BP CAST.THRESH	0.00	GROUP BP CAST.THRESH	0.00	GROUP BP CAST.THRESH	0.00
WRITE FAILED-NO STOR	0.00	WRITE FAILED-NO STOR	0.00	WRITE FAILED-NO STOR	0.00

GROUP TOTAL	QUANTITY
GROUP BP R/W RATIO (%)	21.36
GBP-DEPENDENT GETPAGES	1363.00
SYN.READ(XI)-DATA RETURNED	91.00
SYN.READ(XI)-NO DATA RETURN	445.00
SYN.READ(NF)-DATA RETURNED	3.00
SYN.READ(NF)-NO DATA RETURN	20.00

CLEAN PAGES SYN.WRTN	0.00
CLEAN PAGES ASYN.WRT	0.00
CHANGED PGS SYN.WRTN	440.00
CHANGED PGS ASYN.WRT	0.00
REG.PG LIST (RPL) RQ	3.00
CHANGED PGS READ RPL	0.00
PAGES CASTOUT	337.00
WRITE AND REGISTER	308.00
WRITE AND REGISTER MULT	48.00
READ FOR CASTOUT MULT	34.00
PAGES WRITE & REG MULT	132.00
EXPLICIT X-INVALIDATIONS	0.00
CASTOUT CLASS THRESH	1.00
GROUP BP CAST.THRESH	0.00
WRITE FAILED-NO STOR	0.00

Exception processing

Exception processing identifies Statistics report, trace, and File data set entries with fields that contain values outside the thresholds specified in the Exception Threshold data set. When exception processing is active, the data to be reported is checked against these thresholds.

There are two threshold types that you can set: *warning* and *problem*. A warning message is printed if a value is outside the first threshold and a problem message is printed if a value is outside the second threshold.

When exception processing is requested for a File data set, only delta records in exception status are included in the output data set.

Statistics exception reports, traces, and File data sets are identical to the usual Statistics reports, traces, and File data sets, except that they only contain interval records and delta records that have at least one field in exception status.

When delta records and interval records are processed during exception processing, values defined in the Exception Threshold data set are compared with the corresponding values in a delta or interval record. If any field is in exception status, the following occurs:

- If the JCL contains a valid DD statement for EXTRCDD1 or EXFILDD1, the identity of the delta or interval record (timestamp and subsystem identification) and the fields in exception status are logged in the exception logs.
- If you specified EXCEPTION with the REPORT, TRACE, or FILE subcommand, the interval or delta record is formatted and printed (or filed in case of the FILE subcommand) in the requested layout followed by the Exception Messages block of the report. An example is shown in “Statistics Exception Messages report block” on page 203. The Exception Messages block of the report identifies the fields in exception status.

Note:

- The type of layout used has no effect on exception processing or on the content of the Exception Messages block of the report. All fields in exception status are reported. The layout used only affects the level of detail in the formatted interval or delta record. This means, the exception report can contain fields that have not been included in the corresponding trace or report.
- For group-scope reports, exceptions are checked only against the interval records that are aggregated across a data sharing group. However, member-specific statistics are shown wherever exceptions are found for the entire group.

The fields that caused the entry to be in exception status are identified along with the appropriate warning or problem message and are printed in the Exception Messages block of the report or trace.

“Statistics Exception Messages report block” shows an example of an Exception Messages block . It is printed after each interval or delta record that is found to be in exception status, provided that EXCEPTION was specified in the REPORT or TRACE subcommand.

Statistics Exception Messages report block

Here is an example of a Statistics Exception Messages report block.

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

The following columns are presented in the Exception Messages report block:

TYPE The type of exception. Valid values are WARNING and PROBLEM.

FIELD ID

The name of the field from the Exception Threshold data set.

FIELD DESCRIPTION

The description of the field.

FIELD QUALIFIER

The qualifier of the field. This is either the buffer pool ID, the group buffer pool ID, or the remote location. This field is blank if there is no field qualifier.

BY The basis for the comparison. The following information can be printed in this column:

TOTAL

The threshold is specified as a “total” value.

MINUTE

The threshold is specified as a “by minute” value. The value in the interval or delta record is divided by the number of minutes in the interval or delta before making the comparison.

SECOND

The threshold is specified as a “by second” value. The value in the interval or delta record is divided by the number of seconds in the interval or delta before making the comparison.

COMMIT

The threshold is specified as a “by commit” value. The value in the interval or delta record is divided by the number of Commits in the interval or delta before making the comparison.

THREAD

The threshold is specified as a “by thread” value. The value in the interval or delta record is divided by the number of threads in the interval or delta before making the comparison.

VALUE

The actual field value that is used for the comparison. For total comparisons, the value from the interval or delta record is printed. For “by minute”, “by second”, “by commit”, or “by thread” comparisons, the calculated “by minute”, “by second”, “by commit”, or “by thread” value is printed. The greater than symbol (>) or the less than symbol < is printed between this column and the THRESHOLD column, which indicates whether the value is larger or smaller than the threshold value.

THRESHOLD

The threshold defined in the Exception Threshold data set.

Headers used in Statistics

The headers in Statistics reports and traces are identical, except that INTERVAL FROM and INTERVAL TO in the report header is replaced by ACTUAL FROM in the trace header and the SCOPE field does not apply to Statistics traces.

Statistics report header example

Here is an example of a Statistics report header.

LOCATION: PMODBZ1	OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)	PAGE: 1-1
GROUP: DBZ1	STATISTICS REPORT - LONG	REQUESTED FROM: NOT SPECIFIED
MEMBER: SZ11		TO: NOT SPECIFIED
SUBSYSTEM: SZ11		INTERVAL FROM: 07/18/11 21:30:00.38
DB2 VERSION: V10	SCOPE: MEMBER	TO: 07/18/11 21:53:00.00

The Statistics report and trace headers contain the following information:

LOCATION

The DB2 reporting location. If the DB2 subsystem is installed without a location name, either the DB2 subsystem ID (in non-data sharing environments) or the data sharing group name (in data sharing environments) is shown.

GROUP

The data sharing group that the DB2 subsystem belongs to.

MEMBER

The name of the DB2 data sharing member. This field is not printed for group-scope reports.

SUBSYSTEM

The ID of the DB2 subsystem that generated the data. This field is not printed for group-scope reports.

DB2 VERSION

The version number of the DB2 subsystem that generated the data.

Title - layout

The title of the report or trace and the layout. The layout can be the default layout (LONG or SHORT) or a layout that you have tailored yourself.

SCOPE

The scope of the report, which can be MEMBER or GROUP. A member-scope report shows a group's instrumentation data member by member without merging the data. A group-scope report merges the instrumentation data that belong to the individual members and presents it for the entire group.

Note: All reports in a non-data sharing environment are member-scope reports.

EXCEPTION

Indicates an exception report or trace.

PAGE The page number within the report in the format *lll-nnnnnn*, where *lll* denotes the sequence number of one or multiple locations (see **LOCATION**), and *nnnnnn* denotes the page number within the location.

REQUESTED FROM and TO

The FROM and TO dates and times that were specified in the REPORT or TRACE subcommand. If both FROM and TO dates and times are omitted from the subcommand, the FROM and TO dates and times from the GLOBAL command are printed.

If only the FROM date and time or only the TO date and time has been specified, NOT SPECIFIED is printed for the unspecified value. If FROM and TO are not specified in the subcommand and GLOBAL command, NOT SPECIFIED appears for both the FROM and TO values.

If you specified FROM and TO times without dates in the subcommand and GLOBAL command, ALL DATES is printed together with the specified times.

Note: Any FROM and TO times that are specified in REDUCE can affect the GLOBAL FROM and TO times.

ACTUAL FROM

The date and time of the first DB2 Statistics record that was processed by the trace.

INTERVAL FROM

The start date and time of the first interval record that is covered by the report.

INTERVAL TO

The end date and time of the last interval record that is covered by the report.

How values are reported

OMEGAMON for DB2 PE reports and traces show field values for both delta and interval reporting.

The field values are shown in the following forms:

TOTAL

The value is a total.

QUANTITY

The value is a total.

SECOND

The value in the interval or delta record is divided by the number of seconds in the interval or delta before it is reported.

COMMIT

The value in the interval or delta record is divided by the number of commits in the interval or delta before it is reported.

THREAD

The value in the interval or delta record is divided by the number of threads in the interval or delta before it is reported.

Chapter 19. Exception processing

Exception reporting identifies Accounting and Statistics records that contain fields with values that are outside a desired range. This helps you to recognize performance problems in the DB2 subsystem and in threads.

You define exception thresholds for specific Accounting and Statistics fields in the Exception Threshold data set. If exception processing is active, the DB2 instrumentation data is checked against the threshold values in the Exception Threshold data set. Only records with at least one field value outside the thresholds are reported.

You can set the thresholds in the Exception Threshold data set by using exception profiling or with the help of the example Exception Threshold data set that is delivered with OMEGAMON XE for DB2 PE. Exception profiling can also be used to produce a report with details of the distribution and expected number of exceptions for each field.

Exception reports and traces are available in the Accounting report set and the Statistics report set. Each of these relates separately to Accounting or Statistics data.

In addition, the exception log lists both Accounting and Statistics exception records in the same report, in timestamp order. The contents of the Exception Log File data set are similar to the exception log. The Exception Log File data set can be used by the DB2 load utility.

“Exception processing output types”

Exceptions are reported in various output types.

“Exception Threshold data set” on page 209

The Exception Threshold data set contains the exception thresholds for the Statistics and Accounting exception reports and traces. When exception processing is active, the instrumentation data is checked against these thresholds.

“Exception profiling” on page 210

You can use the exception profiling method to set adequate exception threshold values based on sample DB2 instrumentation data. This method eases the process of getting a suitable Exception Threshold data set that can be used for exception processing.

Exception processing output types

Exceptions are reported in various output types.

This topic describes the exception processing output.

Exception thresholds are specified in the Exception Threshold data set, ddname EXCPTDD. Therefore, the data set information for EXCPTDD must be specified in your JCL if you want to produce any of the listed output types.

“Accounting and Statistics exception reports” on page 208

An exception report is produced if you specify the EXCEPTION subcommand option in the REPORT subcommand.

“Accounting and Statistics exception traces”

You can produce an exception trace by specifying the EXCEPTION subcommand option in the TRACE subcommand.

“Accounting and Statistics Exception File data sets”

You can produce an example file data set by specifying EXCEPTION in the FILE subcommand.

“Exception log data set”

The exception log data set is produced automatically if you specified EXTRCDD1 in the JCL.

“Exception Log File data set”

The Exception Log File data set is written if you define the EXFILDD1 DD statement in the JCL.

Accounting and Statistics exception reports

An exception report is produced if you specify the EXCEPTION subcommand option in the REPORT subcommand.

Both report types contain entries that have at least one value outside the thresholds that you specified in the Exception Threshold data set.

After each report entry, information about the fields in exception status is printed.

Accounting and Statistics exception traces

You can produce an exception trace by specifying the EXCEPTION subcommand option in the TRACE subcommand.

Both trace types show records with fields values outside the thresholds that you specified in the Exception Threshold data set.

After each trace entry, information about the fields in exception status is printed.

Accounting and Statistics Exception File data sets

You can produce an example file data set by specifying EXCEPTION in the FILE subcommand.

The Exception File data set contains records that have at least one value outside the thresholds that you specified in the Exception Threshold data set.

Exception log data set

The exception log data set is produced automatically if you specified EXTRCDD1 in the JCL.

The exception log data set contains a list, in timestamp order, of DB2 Accounting and Statistics records with at least one field outside user-specified thresholds.

Exception Log File data set

The Exception Log File data set is written if you define the EXFILDD1 DD statement in the JCL.

The Exception Log File data set is a sequential data set that can be used with the DB2 load utility.

The Exception Log File data set contains a list of Accounting and Statistics exception records similar to the list in the Exception Log data set.

To prevent the generation of the Exception Log File data set, omit the EXFILDD1 statement from your JCL (the preferred method), or specify DUMMY in the definition.

You can control the amount of data reported in the Exception Log File data set with GLOBAL INCLUDE or GLOBAL EXCLUDE, and FROM and TO.

Allocation values for EXFILDD1 are:

RECFM:

VB

LRECL:

512

BLKSIZE:

Recommended 4 096

Exception Threshold data set

The Exception Threshold data set contains the exception thresholds for the Statistics and Accounting exception reports and traces. When exception processing is active, the instrumentation data is checked against these thresholds.

Threshold values in an Exception Threshold data set can be set or modified with the Exception Threshold data set editor (see “Specifying exceptions using the Exception Threshold data set editor” on page 66) or the exception profiling method (see “Exception profiling” on page 210).

A sample Exception Threshold data set is supplied in data set member RK02DATA(DG0ETV51). The sample contains a selection of exception fields with predefined threshold values and can be used to get started with exception reporting.

Note: Earlier versions of the sample Exception Threshold data set RK02DATA(DG0ETV51) contain entries with asterisks instead of predefined threshold values. Asterisks are intended to mark thresholds that are to be determined by the exception profiling method (described in “Exception profiling” on page 210). If you use these samples for exception reporting without performing exception profiling (which creates a new Exception Threshold data set with asterisks replaced by calculated values), the entries that contain asterisks generate warning messages during exception processing. In other words, earlier samples of the Exception Threshold data set are intended for exception profiling, later samples can be used for exception processing without modification.

If exception processing is started as part of the OMEGAMON Collector startup, where the Exception Threshold data set to be used is determined by the **AUTOEXCPHNAME** startup parameter, the Exception Threshold data set must be a sequential data set. For online monitoring, the Exception Threshold data set can be either a sequential data set or a member of a partitioned data set. If you create a new data set, preallocate it with the following attributes:

RECFM:

VB

LRECL:

≥ 255

BLKSIZE:

6 233 or greater

Note: The sample Exception Threshold data set member RK02DATA(DGOETV51) might have a different record length. When you copy member DGOETV51 to your newly allocated data set, you might get a warning that records are truncated. In this case, you can ignore this warning.

Related reading: For information about specifying and editing thresholds in an existing Exception Threshold data set, refer to “Specifying exceptions using the Exception Threshold data set editor” on page 66.

Related reading: For information about profiling an Exception Threshold data set with a sample of DB2 instrumentation data, refer to “Exception profiling.”

Exception profiling

You can use the exception profiling method to set adequate exception threshold values based on sample DB2 instrumentation data. This method eases the process of getting a suitable Exception Threshold data set that can be used for exception processing.

The basic procedure is to use an existing Exception Threshold data set, mark those fields that you want to be profiled by an asterisk (*), specify generalized warning and problem exception levels, and let the exception profiling batch job do the calculations based on the sample instrumentation data. The result is a new Exception Threshold data set with adequate exception threshold values.

Exception profiling can be repeated by using different sample data, or different field selections, or different generalized exception levels. An Exception Threshold data set can also be modified as described in “Specifying exceptions using the Exception Threshold data set editor” on page 66.

“Data sets involved in exception profiling”

Exception profiling requires several data set specifications.

“Exception profiling dialog” on page 211

This section describes the invocation and the dialog of exception profiling.

“Exception profiling method” on page 213

Each record in the input data set is processed as it would be for normal exception processing. However, instead of checking each field for exception, the value of the field is recorded. When all records have been processed, you can use the statistics in the profile report to determine the thresholds.

“Exception profiling report” on page 214

Use the Exception profiling report to examine the results of exception profiling.

Data sets involved in exception profiling

Exception profiling requires several data set specifications.

The following data sets are essential for exception profiling. You must specify these data sets in the exception profiling dialog.

- The *input data set* contains a sample of DB2 instrumentation data. The data is used to estimate your workload and decide which warning and problem thresholds to assign to the DB2 fields that are to be calculated.

The more representative the data, the more accurate the calculation of the threshold values. The data should contain the type of data that you usually monitor. The input data should also contain a sufficient number of records to allow the profiling to be performed with reasonable confidence. The data should also cover an appropriate span of time.

You can specify any combination of DPMOUT, SME, or GTF data sets that contain DB2 instrumentation data.

- The *input threshold data set* is an Exception Threshold data set with some DB2 fields marked by asterisks (*). The asterisks serve as markers to indicate the DB2 fields for which exception profiling should calculate threshold values. Exception profiling scans the *input threshold data set* for names of DB2 fields that have asterisks (*) assigned instead of threshold values. For these fields new threshold values are calculated based on sample data from the *input data set*. Fields that already have threshold values assigned are not considered and remain unchanged.

In other words, if you want exception profiling to calculate or recalculate certain threshold values, enter asterisks as threshold values for the corresponding fields in the *input threshold data set*.

- The *output threshold data set* is the new Exception Threshold data set. It is basically a copy of the *input threshold data set* (previously existing threshold values are retained) , but previously marked fields obtained new threshold values.

The *output report data set* is optional. If you request a profile report in the exception profiling dialog, this data set will contain a report with details of the distribution and expected number of exceptions for each DB2 field listed in the new Exception Threshold data set.

Exception profiling dialog

This section describes the invocation and the dialog of exception profiling.

To use exception profiling, access the Interactive Report Facility (IRF) from the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS main menu and select option 6, (Exception profiling). The Exception Profiling panel is displayed, as shown in Figure 42 on page 212.

Use the Exception Profiling panel to specify the profiling criteria required to calculate the threshold values and the required and optional data sets.

DGOFEPO0
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=Incl excl
F12=Cancel

Figure 42. Exception Profiling panel

In this panel:

- Specify the percentage of input data that you want to be flagged as warnings (for example 5.00%) and the percentage of input data that you want to be flagged as problems (for example 2.50%). The percentages can be as small as 0.01%.
- Choose whether you want to produce a profile report. The profile report documents the results of exception profiling (the expected number of exceptions for various thresholds).
- Specify the name of the input data set that contains DB2 instrumentation data from your DB2 subsystem. It can be a GTF, SMF, or DPMOUT data set.
If you want to use several input data sets, you can concatenate them by editing the generated job stream by using option 2, Edit the generated job stream, in the Job Processing Selections panel (see Figure 43 on page 213).
- Specify the name of the input threshold data set that contains entries marked by asterisks (*) for the fields that you want to be profiled.
- Specify the name of the output threshold data set that will contain the results.
- If you requested a profile report, specify the name of the output report data set.

After you have completed the specifications, press Enter to generate the exception profiling job stream. The Job Processing Selections panel is displayed.

DG00JOBM

Job Processing Selections

Command ===>

Update the job statements as required, then select one of the following.

4

1. Browse the generated job stream
2. Edit the generated job stream
3. Store the job stream for future use
4. Submit the job stream for background execution

Job statement information:

//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 43. Submitting the exception profiling job

To submit the job, select option 4, enter your appropriate job statement information, and press Enter. Alternatively, you can browse, edit, or store the job stream for subsequent processing.

The new Exception Threshold data set is created.

Check the profiling report to ensure that the exception thresholds and the number of exceptions are satisfactory. If necessary, you can modify the threshold values in the new Exception Threshold data set, as described in “Specifying exceptions using the Exception Threshold data set editor” on page 66.

Now you can generate exception reports using the new Exception Threshold data set created by the exception profiling function.

Exception profiling method

Each record in the input data set is processed as it would be for normal exception processing. However, instead of checking each field for exception, the value of the field is recorded. When all records have been processed, you can use the statistics in the profile report to determine the thresholds.

The thresholds are set to values that would generate the percentage of warning and problem level exceptions that you specified in the Exception Profiling panel (Figure 42 on page 212).

For example, suppose the field QIESECT is specified in the Exception Threshold data set for location DSNAPC5, with the operator set to greater than (>), and the warning and problem thresholds set to asterisk (*). The profiling criteria are specified as 5% for warning thresholds and 2.5% for problem thresholds in the Exception Profiling panel. If the data set is processed and 320 records are found for field QIESECT with the location DSNAPC5, exception profiling sets the warning threshold to the 17th highest record. This would generate 16 warning exceptions

on average, which is 5% of the records (as you requested). Similarly, the problem threshold is set to the 9th largest record, averaging eight problem exceptions, which is 2.5% of the records.

Exception profiling report

Use the Exception profiling report to examine the results of exception profiling.

Example of an exception profiling report

Here is an example report of exception profiling.

The INPUT FROM and INPUT TO fields show the date and time of the first and last record in the input data set.

The profiling report also shows a table for each of the fields. These tables show the calculated threshold values (SPECIFIED THRESHOLD) and the expected number of exceptions (EXCEPTIONS GENERATED) for various exception percentages. These percentages are in multiples of 0.1, 0.25, 0.5, 1, 1.5, and 2 times the problem exception percentage specified in the Exception Profiling panel. For example, if the problem percentage is 5%, the table columns shown are 0.5%, 1.25%, 2.5%, 5%, 7.5%, and 10%.

Any error messages that are generated during exception processing are shown on the profiling report.

ACTUAL AT: 02/17/13 11:21:58.26 OMEGAMON XE for DB2 Performance Expert (V5R3M0) PAGE: 1
INPUT FROM: 01/02/13 06:02:12.11 EXCEPTION PROFILING REPORT
INPUT TO: 02/15/13 02:30:00.00

WARNING THRESHOLD % 10.00
PROBLEM THRESHOLD % 5.00

LOCATION FIELD ID REQLOC FLD.QUAL SUBSYSTEMID	CONNECT GROUP MEMBER	CORRNAME CORRNMBR	PLANNAME PRMAUTH	PER	BY	OPERATOR	OCCURRENCES	DESCRIPTION
* QBACGET * 103 GETPAGES N/A *	* * * *	* * * *	* * * *	PLAN	TOTAL	>		

	PROBLEM	WARNING	0.50 %	1.25 %	2.50 %	5.00 %	7.50 %	10.00 %
SPECIFIED THRESHOLD	22	21	39	37	37	22	21	21
EXCEPTIONS GENERATED	5	8	1	2	2	5	8	8

* SLRSUSP * 150 TOTAL ALL SUSPENSIONS N/A *	* * * *	* * * *	* * * *	SYSTEM	TOTAL	>		
--	------------------	------------------	------------------	--------	-------	---	--	--

	PROBLEM	WARNING	0.50 %	1.25 %	2.50 %	5.00 %	7.50 %	10.00 %
SPECIFIED THRESHOLD	423603	402603	444603	441603	435603	423603	411603	402603
EXCEPTIONS GENERATED	8	15	1	2	4	8	12	15

* SSCDML * 150 TOTAL SQL DML STATEMENTS N/A *	* * * *	* * * *	* * * *	SYSTEM	TOTAL	>		
--	------------------	------------------	------------------	--------	-------	---	--	--

	PROBLEM	WARNING	0.50 %	1.25 %	2.50 %	5.00 %	7.50 %	10.00 %
SPECIFIED THRESHOLD	1410145	1340145	1480145	1470145	1450145	1410145	1370145	1340145
EXCEPTIONS GENERATED	8	15	1	2	4	8	12	15

EXCEPTION PROFILING REPORT COMPLETE

Chapter 20. Reporting distributed data

OMEGAMON XE for DB2 PE can report activities that are associated with distributed work, where one DB2 subsystem accesses data from other DB2 subsystems.

DB2 uses the Distributed Data Facility (DDF) to allow an application program that is connected to one DB2 system to access data at a remote DB2 system, or any other relational database management system (DBMS) that supports DRDA.

OMEGAMON XE for DB2 PE can produce reports for a single DB2 location, or for a number of different DB2 host locations, when data is present. You can logically concatenate data sets from several locations in your JCL to produce a multilocation report. The data is then reported separately for each subsystem and ordered by location in alphabetic order.

When work is distributed across locations, the report shows activity at the reported location only.

The following information is reported for every location:

- Nondistributed transactions, this means, the allied threads at the reporting location.
- Local activity for distributed transactions that originate at the reporting location, this means, the allied-distributed threads at the reporting location without the corresponding DBATs at other locations.
- Remote activity performed at the reporting location as part of distributed transactions originating at other locations, this means, the DBATs at the reporting location.

Depending on the type of report, further detail information is reported.

- The Statistics report set shows following information in the Global DDF and the DRDA Remote Locations blocks of the report:
 - Distributed Data Facility (DDF) address space CPU times
 - Statistics for each DB2 remote location for DB2 host subsystems (MVS, OS/390, and z/OS)
 - Aggregate Statistics for all locations that use DRDA
 - Other, nonlocation-specific DDF information
- The Accounting report set shows information for specific threads that participate in distributed activity. Information is shown for both requester locations and server locations. The reports show, for example, elapsed times spent at the server site and the number of transactions, conversations, SQL statements, rows, messages, and bytes sent from the requester and received by the server. This information is reported in the Distributed Activity block of the report.
- The SQL Activity report set shows all SQL data within threads. SQL statements are reported at the location where they were executed, both at the requester location and the server location.
- The Explain report set can show information for packages that are bound at a remote location.

If a list of plans to be explained contains a remotely bound package on a DB2 host system, EXPLAIN automatically connects to the server and explains the

remote package. Alternatively, you can specify the server location to which EXPLAIN is to connect and the plans and packages that you want explained.

“Selecting threads”

You can select which threads or thread types are reported by using the INCLUDE and EXCLUDE subcommand options.

“Examples of distributed transactions”

The following examples show the types of distributed transactions and threads that are included in reports and traces.

Selecting threads

You can select which threads or thread types are reported by using the INCLUDE and EXCLUDE subcommand options.

For example, if you want to produce reports that show only DBATs, you can include only threads that have a thread type of DBAT by using INCLUDE THREADTYPE(DBAT).

If you want to report the activity performed at a server location on behalf of a specific location, you can use the INCLUDE and EXCLUDE subcommands with REQLOC to only include data for that location.

If you want to report only distributed data, use EXCLUDE THREADTYPE(ALLIED) to exclude data for allied threads.

Examples of distributed transactions

The following examples show the types of distributed transactions and threads that are included in reports and traces.

“Distributed transaction between DB2 host systems”

This example shows distributed transactions between three DB2 host systems.

“Distributed transaction between DB2 host systems and DB2 systems on other platforms” on page 217

This example shows distributed transactions between DB2 host systems on different platforms.

“Distributed transaction involving a DBAT-distributed thread” on page 218

This example shows distributed transactions that involve a DBAT-distributed thread.

Distributed transaction between DB2 host systems

This example shows distributed transactions between three DB2 host systems.

The following figure shows that all three locations are DB2 host systems that run on MVS, OS/390, or z/OS.

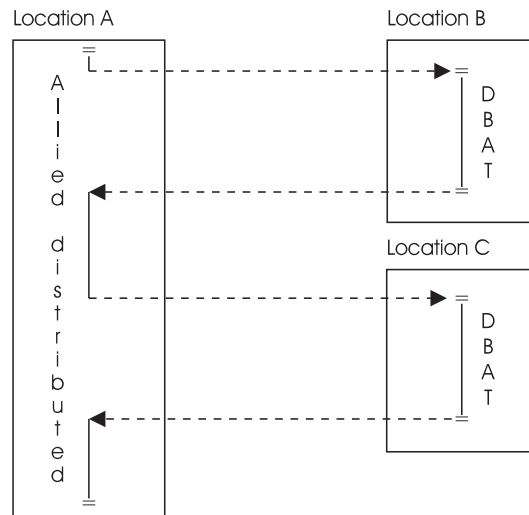


Figure 44. Distributed transaction between DB2 systems

This example shows a thread that originates at location A. Data is requested from location B and location C. Because this transaction is distributed across DB2 systems, it is called an allied-distributed thread at location A. The threads initiated at the remote locations to handle the client requests are called database access threads (DBAT).

If trace data is available, OMEGAMON XE for DB2 PE reports on the activity that is carried out at each location.

Distributed transaction between DB2 host systems and DB2 systems on other platforms

This example shows distributed transactions between DB2 host systems on different platforms.

The following example shows a host DB2 system that runs on MVS, OS/390, or z/OS, and two remote database management systems, such as DB2, that run on different operating systems, such as UNIX.

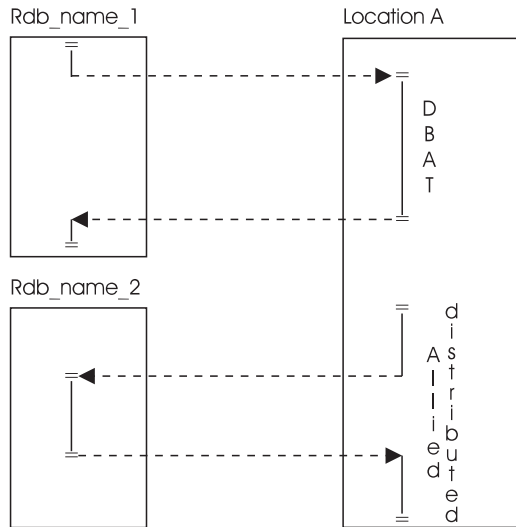


Figure 45. Distributed transaction between DB2 host and other DBMS

The example shows one transaction originated at Rdb_name_1 that requests data at location A and a second transaction originating at location A requesting data from Rdb_name_2.

OMEGAMON XE for DB2 PE reports on the activity that is carried out at Location A only.

Distributed transaction involving a DBAT-distributed thread

This example shows distributed transactions that involve a DBAT-distributed thread.

The following example shows three DB2 host systems.

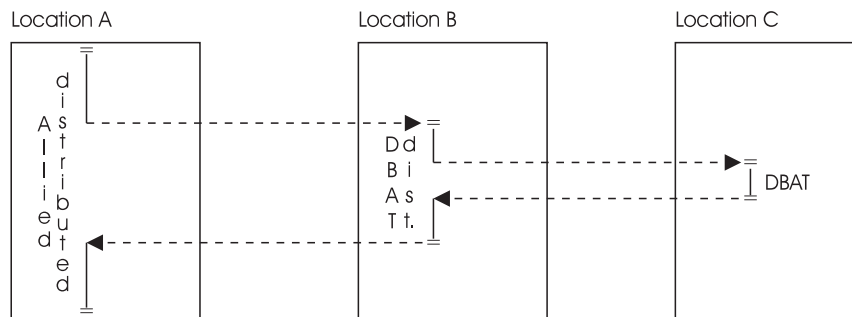


Figure 46. Distributed transaction with a distributed DBAT

A distributed transaction originates at location A, part of the query is passed to location B, which in turn passes some of the query to location C.

When trace data is available for all three locations, a report shows data for the allied-distributed thread at location A, the distributed DBAT at location B, and the DBAT at location C.

Chapter 21. Reporting data sharing information

Data sharing gives individual DB2 subsystems full access to databases that are on shared hard disk drives. The DB2 subsystems that share the data belong to a *data sharing group* and each subsystem is considered a *member* of the group.

In a data sharing environment, you can monitor the performance of entire data sharing groups and individual members of a group. You can do this by generating reports or traces that combine performance information of all members, called *group-scope reports*, or by generating ordinary reports of individual members.

Group-scope reports are available in the Accounting, Locking, Audit, and Statistics report sets.

All report sets provide information about the performance of individual group members. In the report sets where group-scope reports are available, reports for individual members are called *member-scope reports*.

As with all aspects of performance, data sharing is best monitored by using exception processing. You can select exception thresholds for data-sharing-specific fields, and you can specify that the threshold is only checked for a certain group or member.

“Monitoring individual members”

You can monitor various aspects of performance for individual members of a group using any of the OMEGAMON XE for DB2 PE report sets.

“Monitoring entire groups” on page 221

Use group-scope reports to obtain an overall view of the performance of an entire group.

“Collecting input data for group-scope reports” on page 223

To produce group-scope reports, you need input data from all members of a group.

Monitoring individual members

You can monitor various aspects of performance for individual members of a group using any of the OMEGAMON XE for DB2 PE report sets.

“Example of a member-scope Locking report”

Locking reports are helpful in monitoring the locking of page sets.

Example of a member-scope Locking report

Locking reports are helpful in monitoring the locking of page sets.

If you want to monitor deadlocks and timeouts on shared databases, and if you want this information grouped by individual members, generate a Lockout report for every member. Because group-scope reports are available in the Locking report set, this report is called a member-scope report.

Member-scope reports also provide group-scope information because holders and waiters of lockups are shown. Member scope and group scope only influence the summarization of the report.

To generate a member-scope Locking report, specify the following command stream:

```

:
:
LOCKING
  REPORT
    LEVEL (LOCKOUT)
    ORDER(DATABASE-PAGESET)
:

```

Member-scope report is the default, so you are not required to specify the SCOPE subcommand option. To order the report by page set within a database, specify the ORDER subcommand option.

The following example shows a member-scope Locking report for group DSHGRPXX, which has two members, FIRST and SECOND. The report is two pages long because a new page is started when the member that is being reported changes.

Member-scope Locking Lockout report, page 1

The first page of the report shows the locking activity of the threads that have executed in member FIRST.

```

LOCATION: SYDNEY          OMEGAMON XE for DB2 Performance Expert (V5R3M0)          PAGE: 1-1
GROUP: DSHGRPXX          REQUESTED FROM: NOT SPECIFIED
MEMBER: FIRST            TO: NOT SPECIFIED
SUBSYSTEM: DB22          ACTUAL FROM: 05/15/13 12:15:00.21
DB2 VERSION: V10         TO: 05/15/13 13:27:56.09

LOCKING REPORT - LOCKOUT
ORDER: DATABASE-PAGESET
SCOPE: MEMBER

--- L O C K   R E S O U R C E ---
DATABASE  TYPE  NAME  TIMEOUTS  DEADLOCKS  MEMBER  PLANNAME  CONNECT  CORRNAME  CORRNMBR  BLOCKER/
PAGESET                                     HOLDER  WAITER
-----
DBASE9    ROW    PAGE=X'000021'  0          3  FIRST   D3APP01   BATCH   RUNPR01   'BLANK'    2        1
TSPACEXX  ROW    ROW =X'03'      0          3  SECOND  D3APPBB   BATCH   RUNPRBB   'BLANK'    1        1
          INDEXPAGE PAGE=X'002393'  0          1  FIRST   D3APP02   BATCH   RUNPR02   'BLANK'    0        2
          SUBP=X'01'
          ** LOCKOUTS FOR TSPACEXX **          0          4
** TOTAL - DBASE9 **          0          4

DBASE10   DATAPAGE PAGE=X'000055'  0          3  FIRST   D3APP03   BATCH   RUNPR03   'BLANK'    2        1
TSPACEZZ  FIRST   D3APP05   BATCH   RUNPR05   'BLANK'    0        3
          ** LOCKOUTS FOR TSPACEZZ **          0          3
** GRAND TOTAL **          0          7

```

Member-scope Locking Lockout report, page 2

The second page of the report shows the locking activity of the threads that have executed in member SECOND.

```

LOCATION: SYDNEY          OMEGAMON XE for DB2 Performance Expert (V5R3M0)          PAGE: 2-1
GROUP: DSHGRPXX          REQUESTED FROM: NOT SPECIFIED
MEMBER: SECOND           TO: NOT SPECIFIED
SUBSYSTEM: DB22          ACTUAL FROM: 05/15/13 12:15:00.21
DB2 VERSION: V10         TO: 05/15/13 13:27:56.09

LOCKING REPORT - LOCKOUT
ORDER: DATABASE-PAGESET
SCOPE: MEMBER

--- L O C K   R E S O U R C E ---
DATABASE  TYPE  NAME  TIMEOUTS  DEADLOCKS  MEMBER  PLANNAME  CONNECT  CORRNAME  CORRNMBR  HOLDER  WAITER
PAGESET                                     -----
DBASE9    ROW    PAGE=X'000021'  0          4  FIRST   D3APP01   BATCH   RUNPR01   'BLANK'    2        1
TSPACEXX  ROW    ROW =X'03'      0          4  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    ROW =X'03'      0          4  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	D3APPC	BATCH	RUNPRC	'BLANK'	1	0
				SECOND	D3APPD	BATCH	RUNPRD	'BLANK'	0	2
** LOCKOUTS FOR TSPACEZZ	**	0	5							
** GRAND TOTAL **		0	9							

Monitoring entire groups

Use group-scope reports to obtain an overall view of the performance of an entire group.

Group-scope reports are available in the Accounting, Locking, Audit, and Statistics report sets:

- The group-scope Accounting reports merge instrumentation data produced by the individual group members and present it for the entire group.
- The group-scope Locking reports provide a full picture of the locking activity within the entire data sharing group.
- The group-scope Statistics reports summarize group buffer pool and locking information for shared resources for all members. They also show key information, such as total number of threads and commits for an entire group.
- The group-scope Audit reports provide a comprehensive view of the access to shared resources by the users of the members of the group. For example, if you want a summary of users that belong to various members of a group who accessed, or attempted to access, page sets on shared databases, you can generate a group-scope Audit DML access report.

“Example of a group-scope Locking report”

In group-scope reports, events are aggregated by user-defined identifiers within the group, regardless of which member of the group actually generated the events.

“Group-scope Statistics” on page 222

The group-scope Statistics reports show three categories of information summarized by group level.

“Group-scope Accounting” on page 223

Group-scope reports show the instrumentation data aggregated by the OMEGAMON XE for DB2 PE identifiers that you specified and by the individual members.

Example of a group-scope Locking report

In group-scope reports, events are aggregated by user-defined identifiers within the group, regardless of which member of the group actually generated the events.

To generate a group-scope Locking report, specify the following command stream:

```

:
LOCKING
  REPORT
    LEVEL (LOCKOUT)
    SCOPE (GROUP)
:

```

The default order of a group-scope report is DATABASE-PAGESET.

Group-scope Locking Lockout report for DSHGRPXX

The following example shows a group-scope Locking report for the same group, DSHGRPXX, as in the previous member-scope example. This report summarizes

the lockout activity for both members FIRST and SECOND. The information is summarized by database, then page set, and lastly by individual member.

A group total is printed for the entire group when the database that is being monitored changes. GRAND TOTAL shows the timeouts and deadlocks in all databases for the entire group.

LOCATION: SYDNEY GROUP: DSHGRPXX		OMEGAMON XE for DB2 Performance Expert (V5R3M0)					PAGE: 1-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED ACTUAL FROM: 05/15/13 12:15:00.21 TO: 05/15/13 13:27:56.09						
DB2 VERSION: V10		LOCKING REPORT - LOCKOUT ORDER: DATABASE-PAGESET SCOPE: GROUP											
DATABASE PAGESET MEMBER		--- L O C K R E S O U R C E ---					----- A G E N T S -----						
		TYPE	NAME	TIMEOUTS	DEADLOCKS	MEMBER	PLANNAME	CONNECT	CORRNAME	CORRNMBR	HOLDER	WAITER	

DBASE9													
TSPACEXX													
FIRST		ROW	PAGE=X'000021' ROW =X'03'	0	3	FIRST	D3APP01	BATCH	RUNPR01	'BLANK'	2	1	
						SECOND	D3APPBB	BATCH	RUNPRBB	'BLANK'	1	1	
						SECOND	D3APPDD	BATCH	RUNPRDD	'BLANK'	0	2	
		INDEXPAGE	PAGE=X'002393' SUBP=X'01'	0	1	FIRST	D3APP02	BATCH	RUNPR02	'BLANK'	0	1	
		** LOCKOUTS FOR FIRST	**	0	4								
SECOND		ROW	PAGE=X'000021' ROW =X'03'	0	4	FIRST	D3APP01	BATCH	RUNPR01	'BLANK'	2	1	
						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' ROW =X'03'	0	4	FIRST	D3APP01	BATCH	RUNPR01	'BLANK'	1	0	
						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' SUBP=X'03'	0	5	SECOND	D3APPBB	BATCH	RUNPRBB	'BLANK'	1	1	
						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								

Group-scope Statistics

The group-scope Statistics reports show three categories of information summarized by group level.

Highlights

This category presents values such as the total number of threads and commits for the entire group.

Data sharing locks

This category presents locking information for shared resources for all members.

Buffer pool data

This category presents statistics per buffer pool summarized for all members of a group.

All other statistics data is presented in member-scope reports for detailed analysis on member level.

Group-scope Accounting

Group-scope reports show the instrumentation data aggregated by the OMEGAMON XE for DB2 PE identifiers that you specified and by the individual members.

The data is presented by a combination of location and group. Whenever either of these values changes, a new page is started and the page number is initialized.

Collecting input data for group-scope reports

To produce group-scope reports, you need input data from all members of a group.

For regular monitoring, you most likely collect the performance data into SMF data sets. If all DB2 subsystems of the data sharing groups reside in the same OS/390 system, you can collect the data to one SMF data set. If the members of the group reside on different OS/390 systems, you have to concatenate the SMF data sets from all OS/390 systems before you can use the data as input for group-scope reports.

Similarly, if you have collected instrumentation data for the individual members in separate data sets (GTF data sets or data sets created by collected report data), you can concatenate these data sets to generate group-scope reports.

Chapter 22. Streamlining OMEGAMON XE for DB2 PE processing

Streamlining is the process of asking only for information in reports and traces that you really need.

OMEGAMON XE for DB2 PE is a comprehensive reporting tool, but for daily monitoring of DB2 you need a very limited amount of information. If you request a detailed report by using all input data that was gathered for a long time, OMEGAMON XE for DB2 PE processing takes up a lot of system resources. The result is pages of information that you are probably not interested in.

To avoid unnecessary performance overhead and to save time:

- Filter the input data, preferably by using the GLOBAL command.
- Disable OMEGAMON XE for DB2 PE internal sort if appropriate.
- Consider carefully how detailed a report you need.
- Define groups for identifiers that you want reported as a single entry.
- Use lists to simplify your command stream.
- Specify a REDUCE INTERVAL only if you want to report by intervals or produce several reports with different time spans.
- Define exception thresholds only for fields that you are interested in.
- Specify DPMOUT or keep a Save data set only if you are sure that you want to report the data again.
- Limit the number of SQL statements you want explained.

“Filter data” on page 226

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.

“Suppressing internal sort” on page 230

When you request Accounting functions only, it is often possible to avoid the OMEGAMON XE for DB2 PE internal sort of the input data. This reduces the size of the sort work files that must be allocated and reduces the processing time.

“Choose the right level of detail” on page 230

Before you produce a report or a trace, consider how much detail you need. Use the LAYOUT subcommand option of the Accounting and Statistics reports and traces to control the amount of data to be produced.

“Group data” on page 231

Use the GROUP command to define a set of OMEGAMON XE for DB2 PE identifier values that can be used when you request certain reports.

“Use lists” on page 231

Use the LIST command to define a list of values for an OMEGAMON XE for DB2 PE identifier that can be used in INCLUDE and EXCLUDE, instead of individually entering each member. The members of the list are treated as if they were entered individually.

“Be careful with INTERVAL” on page 232

If you want to use OMEGAMON XE for DB2 PE for trend analysis, you probably want to report data by interval. To do this, first reduce the input data

to your Accounting and Statistics reports by using an interval other than the default zero. Then order the report by intervals.

“Specify only relevant exception thresholds” on page 232

Exception processing is the most effective way to identify system performance problems. You can set exception thresholds for virtually all Accounting and Statistics fields. However, carefully consider the fields for which to specify exception thresholds. The more fields you specify, the greater the effects on processing.

“Do you need a DPMOUT data set?” on page 233

You can keep the preprocessed input data in the output data set DPMOUT. The DPMOUT data set can be used as input to OMEGAMON XE for DB2 PE.

“Do you want to save reduced data?” on page 233

If you want to keep historical Accounting and Statistics data about DB2 performance, consider using REDUCE and SAVE.

“Limit statements to be explained” on page 233

When you produce Explain reports, limit the number of SQL statements to be examined.

Filter data

You can limit the amount of data to be processed by filtering the input data. You can specify filters in the GLOBAL command or in the REDUCE, REPORT, TRACE, or FILE subcommands.

You should specify the filters in GLOBAL, because only the data that passes through the GLOBAL filters is processed further. Ensure that you do not exclude records that are needed in subsequent processing.

“FROM and TO”

The simplest filter is the start and end time of the data to be reported.

“INCLUDE and EXCLUDE” on page 228

Another way to filter data is to include data only for particular OMEGAMON XE for DB2 PE identifier values, for example, user IDs or plans.

FROM and TO

The simplest filter is the start and end time of the data to be reported.

Specify the start and end times by using the FROM and TO subcommand options.

“Example of reporting peak hour performance of a day”

This example shows the use of FROM and TO within GLOBAL to report a specified date and time frame.

“Example of reporting peak hour performance during a week” on page 227

This example shows the use of FROM and TO within GLOBAL to report a specified time frame every day.

“Example of generating an additional locking report” on page 227

This example shows the use of FROM and TO for two purposes in the same command stream.

Example of reporting peak hour performance of a day

This example shows the use of FROM and TO within GLOBAL to report a specified date and time frame.

If you want to monitor the performance of your system only during peak hours and you want to produce the default versions of both, an Accounting report and a Statistics trace, you can specify:

```

:
GLOBAL
    FROM (05/15/13 ,08:30)
    TO   (05/15/13 ,17:00)
ACCOUNTING
    REPORT
STATISTICS
    TRACE
:

```

Both, the Accounting report and Statistics trace show information from 8:30 a.m. to 5 p.m. for the specified day.

Example of reporting peak hour performance during a week

This example shows the use of FROM and TO within GLOBAL to report a specified time frame every day.

If the input data set contains data of at least a week, you can generate reports that show the performance of your system during peak hours for the whole week by specifying:

```

:
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.

Example of generating an additional locking report

This example shows the use of FROM and TO for two purposes in the same command stream.

If you want to generate a Locking report for a day in the same job step, specify:

```

:
GLOBAL
    FROM (,08:30)
    TO   (,17:00)
ACCOUNTING
    REPORT
STATISTICS
    TRACE
LOCKING
    REPORT
        FROM (05/17/13 )
        TO   (05/17/13 )
:

```

The Locking report contains data only from 8:30 a.m. to 5 p.m. for that day because no other data has passed the GLOBAL filtering.

INCLUDE and EXCLUDE

Another way to filter data is to include data only for particular OMEGAMON XE for DB2 PE identifier values, for example, user IDs or plans.

You can do this by using the INCLUDE and EXCLUDE subcommand options.

“OMEGAMON XE for DB2 PE identifiers”

The identifiers describe the objects OMEGAMON XE for DB2 PE reports on.

“Example of reporting the performance of a specific location” on page 229

This example shows how only a single location identifier is included.

“Example of suppressing a specific connection type” on page 229

This example shows how a single location is included and a single connection type is excluded.

“Example of reporting about a specific user ID and plan” on page 229

This example shows how a single authorization identifier and a single plan name is included.

OMEGAMON XE for DB2 PE identifiers

The identifiers describe the objects OMEGAMON XE for DB2 PE reports on.

The most commonly used OMEGAMON XE for DB2 PE identifiers describe:

- **The location**

The LOCATION identifier is the name of the DB2 subsystem. If the input data contains data from several locations, you can include data only for those locations that you are interested in.

- **The user**

OMEGAMON XE for DB2 PE uses two identifiers for the user ID. The first is the value of the authorization ID at the time of connection to DB2 (ORIGAUTH). The second is the authorization ID set at signon or identify (PRIMAUTH). For more information, see “Comparing original authorization IDs with primary authorization IDs” on page 286.

For SQL requests from a client, the user ID of the user at the workstation is a possible identifier as well.

- **The plan**

Use the PLANNAME identifier to select specific plans. Examples of plan names are DSNUTIL for utility, DSNBIND for bind activity, and the application plan name for CICS and IMS.

- **The package**

Use the PACKAGE or MAINPACK identifiers to select plans and packages in the Accounting report set. Use PACKAGE to define specific packages, regardless of the plan to which they belong. Use MAINPACK to define plans that contain a specific package. For more information, see “Defining the MAINPACK identifier” on page 252.

- **The connection to DB2**

You can select data for specific connections to DB2.

The connection ID (CONNECT) identifies the address space that interfaces with DB2. It can be, for example, the CICS or IMS ID.

The connection type (CONNTYPE) identifies the type of connection for a thread. It can be, for example, CICS, IMS-BMP, IMS-MPP, IMS-CNTL, or IMS-TBMP.

- **The correlation data**

The correlation identifier identifies the DB2 task together with the connection ID. It is composed of two parts: the correlation name (CORRNAME) and the

correlation number (CORRNMBR). This identifier can be very useful. For example, for CICS threads the correlation identifier contains the transaction ID. For more information, see “Correlation ID translation” on page 251.

- **Distributed activity**

If you report data for distributed processing, you can select the locations that request the work (REQLOC) and the type of the threads (THREADTYPE) to be included, for example allied threads or DBATs.

In addition, the following identifiers of a client are supported to include or exclude related data:

- The end user’s user ID at the workstation (ENDUSER)
- The end user’s transaction name at the workstation (TRANSACT)
- The end user’s workstation name (WSNAME)

Example of reporting the performance of a specific location

This example shows how only a single location identifier is included.

If the input data contains data from more than one location, but you only want to inspect the performance of location USIBMSNEWY11, specify:

```
⋮
GLOBAL
    INCLUDE (LOCATION(USIBMSNEWY11))
⋮
```

Only data for location USIBMSNEWY11 is processed.

Example of suppressing a specific connection type

This example shows how a single location is included and a single connection type is excluded.

If you are not interested in the CICS activity for that location, specify:

```
⋮
GLOBAL
    INCLUDE (LOCATION(USIBMSNEWY11))
    EXCLUDE (CONNTYPE(CICS))
⋮
```

Only location USIBMSNEWY11 is reported and all connections except CICS are reported.

Example of reporting about a specific user ID and plan

This example shows how a single authorization identifier and a single plan name is included.

If you suspect that authorization identifier USERID01 using plan NEWACC is causing a performance problem, you can report only data that belong to that user ID and plan. To do that, specify:

```
⋮
GLOBAL
    INCLUDE (AUTHID(USERID01)
            PLANNAME(NEWACC))
⋮
```

Only data with authorization ID USERID01 and plan NEWACC is reported.

Suppressing internal sort

When you request Accounting functions only, it is often possible to avoid the OMEGAMON XE for DB2 PE internal sort of the input data. This reduces the size of the sort work files that must be allocated and reduces the processing time.

The PRESORTED option of the GLOBAL command controls the internal sort. It has the following values:

NO Do not disable the sort. This is the default.

ENFORCE

Disable the sort, but terminate processing if out-of-sequence records are present. Use this option only if the input data set has been sorted, for example, when you reprocess data from the DPMOUT data set.

ACCEPT

Disable the sort and accept out-of-sequence records. By using this option it is possible to create complete and accurate Accounting reports (including records from multiple locations) from data sets that have not been sorted, for example SMF or GTF. The following limitations apply with this option:

- Some checking of the input data is not performed. Ensure that the concatenation of two or more data sets does not result in duplicate records, for example, if trace data was collected in two data sets during the same period.
- Only one location is reported per trace. To report more locations, specify multiple TRACE subcommands with different INCLUDE subcommand options for each location.
- Trace entries might not be printed in time sequence. If a trace contains entries that are out of sequence, a message that shows the number is displayed at the end.
- If INTERVAL(0) is in effect (the default), the interval times shown in a report heading might not be accurate. Ordering reports by interval should be avoided when the interval is zero.

For example, to produce an Accounting report without sorting the input data, specify:

```
⋮  
GLOBAL  
    PRESORTED(ACCEPT)  
  
ACCOUNTING  
⋮
```

The resulting Accounting report shows Accounting data for all locations in the input data set, without performing an internal sort.

Choose the right level of detail

Before you produce a report or a trace, consider how much detail you need. Use the LAYOUT subcommand option of the Accounting and Statistics reports and traces to control the amount of data to be produced.

In most situations the default layouts, which are short versions of reports and traces, provide enough detail for monitoring and problem determination. Do not use the most comprehensive layouts that show all possible fields unless you need this information.

For Accounting the LAYOUT subcommand option ACCEL provides detailed thread-related Accelerator activity data.

If the use of commands, subcommands, and subcommand options does not provide sufficient control over the contents of reports and traces, you can use the User-Tailored Reporting (UTR) feature to create and tailor your own report and trace layouts. With UTR you can add, remove, and change individual fields and entire report blocks to control the volume, contents, and layouts of your reports and traces. For more information, see “Tailoring report layouts” on page 236.

Group data

Use the GROUP command to define a set of OMEGAMON XE for DB2 PE identifier values that can be used when you request certain reports.

The information for the set is reported as a single entry in the reports.

In GROUP processing the data for all members of the set is consolidated into one record. This improves OMEGAMON XE for DB2 PE performance because fewer records must be processed.

Sets are also useful for reporting purposes such as if you want to report data for an entire department instead of every individual person.

This is how you specify sets. Assume that your sales department consists of three users, USER001, USER002, and USER003. You want to produce an Accounting report that shows performance data for that department. You can enter:

```
GROUP          (PRMAUTH(SALES(USER001,
                                USER002,
                                USER003)))

ACCOUNTING
  REDUCE
    INCLUDE (PRMAUTH(G(SALES)))
  REPORT
    ORDER  (PRMAUTH)
  :
```

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 PRMAUTH
- PLANNAME
- CONNECT and CONNTYPE
- CORRNAME and CORRNMBR

Use lists

Use the LIST command to define a list of values for an OMEGAMON XE for DB2 PE identifier that can be used in INCLUDE and EXCLUDE, instead of individually entering each member. The members of the list are treated as if they were entered individually.

LIST processing does not affect OMEGAMON XE for DB2 PE performance, but can make it easier to specify and read your command stream.

Assume that you want to see how some plans that are used by the sales department affect performance. To do that, produce an Accounting report and an SQL Activity trace. If you specify a list for all plan name values, as in the following example, you can use the list name (SALES) in all commands in that job step.

```
⋮
LIST          (PLANNAME(SALES(PLAN001,
                           PLAN002,
                           PLAN003,
                           PLAN004,
                           PLAN005,
                           PLAN006,
                           PLAN007,
                           PLAN008)))

ACCOUNTING
  REPORT
    INCLUDE (PLANNAME(L(SALES)))
    ORDER (PLANNAME)
SQLACTIVITY
  TRACE
    INCLUDE (PLANNAME(L(SALES)))
⋮
```

Both, the Accounting report and the SQL Activity trace show information for all individual plans that are specified in the list.

You can use LIST with all report sets. You can specify lists for any OMEGAMON XE for DB2 PE identifier values that are allowed in that report set.

Be careful with INTERVAL

If you want to use OMEGAMON XE for DB2 PE for trend analysis, you probably want to report data by interval. To do this, first reduce the input data to your Accounting and Statistics reports by using an interval other than the default zero. Then order the report by intervals.

For example, if you want to report the data at daily intervals, specify INTERVAL (1440) in the REDUCE subcommand and ORDER(INTERVAL) in the REPORT subcommand.

Interval processing affects the performance of the job. If you do not intend to produce reports by interval, accept the default (0) for INTERVAL, which means that no interval processing is performed.

Specify only relevant exception thresholds

Exception processing is the most effective way to identify system performance problems. You can set exception thresholds for virtually all Accounting and Statistics fields. However, carefully consider the fields for which to specify exception thresholds. The more fields you specify, the greater the effects on processing.

Specify exception thresholds only for those fields that you believe will signal poor performance in your environment. For information about how to generate

exception reports and for a list of threshold fields that you should use, see Chapter 8, “Using Statistics and Accounting reports to identify exceptions,” on page 65.

Do you need a DPMOUT data set?

You can keep the preprocessed input data in the output data set DPMOUT. The DPMOUT data set can be used as input to OMEGAMON XE for DB2 PE.

If you do not specify DPMOUTDD, only the records required for the current job step are processed, which improves system performance.

Do you want to save reduced data?

If you want to keep historical Accounting and Statistics data about DB2 performance, consider using REDUCE and SAVE.

This processing has advantages and disadvantages. Reducing and saving data uses a considerable amount of system resources, but the resulting Save data set is much smaller than the original input data set. The size of the resulting Save data set varies with the specified reduction interval and the type of environment the data is from (for example, the number of different users and plans present in the input data).

You can produce reports from the reduced and saved data by using the RESTORE command. Remember that you cannot produce traces from reduced data.

For more information refer to “Saving reduced data” on page 157.

Limit statements to be explained

When you produce Explain reports, limit the number of SQL statements to be examined.

There are various ways you can limit the output. You can specify only the plans or packages that you are interested in, or you can use the LIMIT, PACKLIMIT, and PACKAGES NO options.

To reduce attachment costs, it is always best to group the subsystems you are reporting.

Chapter 23. Customizing OMEGAMON XE for DB2 PE functions

Customizing is the process of tailoring OMEGAMON XE for DB2 PE functions for specific needs, for example, specifying thresholds or modifying report layouts.

There are some OMEGAMON XE for DB2 PE functions that you can customize to meet your particular needs. The things you can tailor are:

- Exception thresholds

Exception reporting identifies DB2 threads and Statistics intervals with fields values outside defined limits. You can either specify the thresholds yourself (for more information see Chapter 8, “Using Statistics and Accounting reports to identify exceptions,” on page 65) or you can let OMEGAMON XE for DB2 PE fill these values (for more information see “Exception profiling” on page 210).

- Report layouts

Several model layouts for Accounting and Statistics reports are provided with the product. If none of them meets your needs, you can tailor them or create your own report layouts.

- Time zone specifications

You can modify the times used in reporting if the CPU clock of your z/OS system is not set to the local time or if you want to report data from two or more systems that have different CPU clock settings.

- The correlation name and number

OMEGAMON XE for DB2 PE provides a default translation of the DB2 correlation ID for most environments. You can change the way the DB2 correlation ID is translated into correlation name and correlation number.

- MAINPACK identifier

MAINPACK is an OMEGAMON XE for DB2 PE identifier that you can use to identify a plan by the first or the last package within the plan. You can also define whether to use the package ID, the collection ID, or the location from the package name to identify the main package.

- Exception field descriptors

You can modify exception field descriptors by using the ISPF editor.

Changes to these functions are recorded in the DPMPARMS data set. The data set needs to be allocated. You can create multiple DPMPARMS data sets and you can concatenate several DPMPARMS data sets as required.

“Tailoring report layouts” on page 236

OMEGAMON XE for DB2 PE supplies several Accounting and Statistics model reports and traces. You can create your own layouts and you can tailor layouts with the User-Tailored Reporting feature (UTR).

“Specifying time zones” on page 245

The TIMEZONE option of the GLOBAL command provides a means of adjusting the times of the data to be reported.

“Correlation ID translation” on page 251

The correlation ID is a DB2 field that identifies the task executed by DB2.

“Defining the MAINPACK identifier” on page 252

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. “Modifying exception field descriptors” on page 258
 You can change the descriptive labels for exception fields as they are printed for the exception records in batch reports.

Tailoring report layouts

OMEGAMON XE for DB2 PE supplies several Accounting and Statistics model reports and traces. You can create your own layouts and you can tailor layouts with the User-Tailored Reporting feature (UTR).

With UTR you can:

- Add entire blocks and individual fields to an existing layout, for example, to include some additional fields or entire blocks of related fields in a report or trace.
- Remove entire blocks and individual fields from an existing layout, for example, to exclude some fields or entire blocks of related fields that are not of interest to you from a report or trace.
- Change the relative positions of blocks and fields in an existing layout, for example, to rearrange blocks and fields that are of particular interest.
- Change block and field labels, for example, to abbreviate labels to keep reports compact, or to spell out block and field names, or to use acronyms.

“Example of tailoring an Accounting report”

This example describes how to add a block of fields to an existing report layout and how to replace some of the existing fields with new fields.

Example of tailoring an Accounting report

This example describes how to add a block of fields to an existing report layout and how to replace some of the existing fields with new fields.

Suppose that the standard layout of the short Accounting report (supplied as Accounting report layout SHORT) does not provide the information that you need to monitor your DB2 installation and its specific workload.

Accounting report - standard layout

Here is an example of the standard layout for Accounting reports.

LOCATION: STLEC1		OMEGAMON XE for DB2 Performance Expert (V5R3M0)							PAGE: 1-1						
GROUP: DSNCAT		ACCOUNTING REPORT - SHORT							REQUESTED FROM: NOT SPECIFIED						
MEMBER: SSDQ		ORDER: PRIMAUTH-PLANNAME							TO: NOT SPECIFIED						
SUBSYSTEM: SSDQ		SCOPE: MEMBER							INTERVAL FROM: 05/15/13 20:18:00.23						
DB2 VERSION: V11									TO: 05/15/13 20:48:38.68						
PRIMAUTH		#OCCURS	#ROLLBK	SELECTS	INSERTS	UPDATES	DELETES	CLASS1	EL.TIME	CLASS2	EL.TIME	GETPAGES	SYN.READ	LOCK	SUS
PLANNAME		#DISTR	#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
:															

Accounting report - tailored layout

Assume that you want to include more buffer pool information in your layout, as in the following tailored report layout.

The tailored layout has been named BUFFER and contains buffer pool information in addition to General Accounting data. The buffer pool fields in the General Accounting block have been replaced with other fields to avoid duplicate information. In this example, class 3 suspension times and events are shown.

LOCATION: STLEC1		OMEGAMON XE for DB2 Performance Expert (V5R3M0)						PAGE: 1-1						
GROUP: DSNCAT		ACCOUNTING REPORT - BUFFER						REQUESTED FROM: NOT SPECIFIED						
MEMBER: SSDQ		ORDER: PRIMAUTH-PLANNAME						TO: NOT SPECIFIED						
SUBSYSTEM: SSDQ		SCOPE: MEMBER						INTERVAL FROM: 05/15/13 20:18:00.23						
DB2 VERSION: V11								TO: 05/15/13 20:48:38.68						
PRIMAUTH PLANNAME		#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	CL3 CL3	SUSP SUSP.TIME	LOCK #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	#SYNWRT						
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	#SYNWRT						
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						

The following steps describe how to produce the tailored layout.

“Selecting a model layout”

These steps describe how to select an existing report layout as the basis for a new layout that is to contain tailored information.

“Adding a block” on page 239

These steps describe how to add a report block to a selected report layout that is to contain tailored information.

“Replacing fields” on page 241

In this example, the duplicate fields in the GENERAL Accounting report block are replaced with class 3 suspensions and class 3 suspension times.

“Displaying the layout” on page 243

These steps describe how to display the modified layout.

“Saving the layout” on page 244

These steps describe how to save the modified layout after you have completed the changes to your layout.

“Using the layout” on page 245

These steps describe how to use the modified layout.

Selecting a model layout

These steps describe how to select an existing report layout as the basis for a new layout that is to contain tailored information.

About this task

1. From the IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS main menu, select option 5 (Customize report and trace layouts).

The User-Tailored Reporting Layout Generation panel is displayed.

```

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
'DG0710.MYMPARMS'_____

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

```

Figure 47. User-Tailored Reporting - selecting an Accounting report as model layout

- For this example, select option 1 (Accounting report) from the menu, enter the name of the DPMPARMS data set where the tailored layout is to be stored, and press Enter.

The UTR Layout Selection panel is displayed.

```

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 48. User-Tailored Reporting - selecting a short Accounting report as model layout

- For this example, select layout SHORT as the basis for your layout by entering a selection character (/) in the input field next to the layout name. Always choose the model that is most similar to the layout you want.

The UTR Block Selection panel for the Accounting Report SHORT layout is displayed.

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 49. User-Tailored Reporting - report block selection and ordering

The Order column shows which blocks are included in the selected layout. In this example, the blocks labeled GENERAL, PACKAGE GENERAL, and DISTRIBUTED ACTIVITY are included. The numbers in the Order column indicate the sequence in which the blocks are printed. Fields that have not been selected are numbered 99999.

Adding a block

These steps describe how to add a report block to a selected report layout that is to contain tailored information.

About this task

In this example, the BUFFER POOL ACTIVITY block is added to the selected layout.

1. In the UTR Block Selection panel, use F8 to scroll to the label BUFFER POOL ACTIVITY. Notice that there are two blocks with this label. Press F5 to display a brief description of the block's content, as shown in the following figure. In this example, the column form is chosen.

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 50. User-Tailored Reporting - report block contents

- Enter a number in the Order column next to the block label to determine the relative position of the added block to the existing blocks.
In this example, a number between 10 and 20 determines that the BUFFER POOL ACTIVITY block is printed after the GENERAL block and before the PACKAGE GENERAL block. A number less than 10 would add the BUFFER POOL ACTIVITY block before the GENERAL block.
- To see which fields can be included in the added block, enter a selection character (/) in the Modify column next to the block label.
The UTR Field Selection panel is displayed.

DG0BFLD4

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 (QBACSI0)	8
80	BUF.UPDT	Average number of buffer updates (QBACSW)	8
90	#SYNWRT		

Command ==>

F1=Help
 F2=Split
 F3=Exit
 F5=Extend
 F6=Qualify
 F7=Up
 F8=Down
 F9=Swap
 F10=Config
 F11=Filler
 F12=Cancel

Figure 51. User-Tailored Reporting - report block field selection and ordering

Similar to the UTR Block Selection panel, use F7 (Up) and F8 (Down) to scroll through the list of fields and F5 (Extend) to display extended field descriptions. The Order column determines the order of the fields within a block. A value of 99999 means that a field is not selected.

- For this example, assume that all fields of the BUFFER POOL ACTIVITY block are selected by default. Because no fields need to be added or deleted, press F12 (Cancel) to return to the UTR Block Selection panel.

Replacing fields

In this example, the duplicate fields in the GENERAL Accounting report block are replaced with class 3 suspensions and class 3 suspension times.

About this task

You might have noticed that the Buffer Pool block contains some fields (GETPAGES, SYN.READ, BUF.UPDT, and TOT.PREF) that are also in the General Accounting block (see “Accounting report - standard layout” on page 236). To modify the fields in the General Accounting block, perform the following steps:

- In the UTR Block Selection panel, type a forward slash (/) in the Modify column next to GENERAL and press Enter.

The UTR Field Selection panel is displayed.

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 (QBACSW)	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	DESCRIBE	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 52. User-Tailored Reporting - deleting fields

- In the UTR Field Selection panel, use F8 (Down) to find the fields that you do not want in the General Accounting block. Overtyping the numbers (170, 180, 190, and 200) in front of the fields with blanks and pressing Enter (to reorder the remaining fields).
- Use F7 (Up) and F8 (Down) to find the fields that 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 53. User-Tailored Reporting - adding fields

- For this example, enter any number from 160 to 210 in the Order column in front of the fields that you want to add and press Enter.
- Scroll back to see that the newly inserted fields appear at the position that you intended.

You do not have to delete existing fields when you add new fields. You can insert new fields between existing fields or place new fields before or after existing fields. Also, you can delete fields without replacing them with new fields. When the layout is generated the gaps are automatically filled in.

There is no limitation to how many fields or blocks can be selected in a layout. However, carefully consider what information you really need to prevent your reports from getting too long.

Displaying the layout

These steps describe how to display the modified layout.

About this task

- Press F3 (Exit) to return to the UTR Block Selection panel.
- Press F6 (Browse) to display the new layout.
A sample report using the modified layout is displayed.

```
DGOFBRWS SYS94326.T153556.RA000.XXASP36.R0000110 --- Line 00000000 Col 047 126
***** Top of Data *****
OMEGAMON XE for DB2 Performance Expert (V5R3M0) 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 #SYNWRT|

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

Figure 54. User-Tailored Reporting - browsing a sample layout

- 3. You might have to scroll right (F11) and down (F8) to see the entire report. Figure 54 shows the right-hand side of the sample report.

Saving the layout

These steps describe how to save the modified layout after you have completed the changes to your layout.

- 1. Use F3 (Exit) as required to return to the UTR Block Selection panel.
- 2. If you do not want to save the layout, press F12 (Cancel) to leave the UTR Block Selection panel.
- 3. To save the layout, press F3 (Exit).

The UTR Save Layout Details panel is displayed.

```
DG0BP5AV 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 55. User-Tailored Reporting - saving the layout

4. Type a new name for the new layout in the Layout field and a descriptive text in the Description field.
 To keep the existing layout SHORT, and create an additional layout, give the layout a new name by typing over the existing name. You can also type over the existing description.
 You can save the modified layout with the name of the layout you based it on. However, if you do this, the documentation does not reflect your new layout. If you pressed Enter, you would save the modified layout with the name SHORT. In this example, the layout is named BUFFER and the description is changed.
5. Specify whether you want each new record on the report or trace to start on a new page.
 If your layout contains several blocks of information, it is often clearest to have every entry start on a new page.
 If your layout only contains a few blocks of information, several entries most likely fit on the page. Also, even if your layout is long, but you intend to use it with options, such as ORDER(PACKAGE), that limit the amount of information produced, you probably want more than one entry per page.
6. Press Enter.
 The Accounting report layout BUFFER is saved into your DPMPARMS data set.

The layout change is now complete. The UTR Layout Selection panel is displayed. Notice that the layout that you just created is shown in the list of layouts. To exit from this panel, press F3.

Using the layout

These steps describe how to use the modified layout.

About this task

1. Specify the name of your DPMPARMS data set in the JCL.
2. You can now run Accounting reports using the new layout.

```

      :
      ACCOUNTING
      REPORT
      LAYOUT (BUFFER)
      :

```

Specifying time zones

The TIMEZONE option of the GLOBAL command provides a means of adjusting the times of the data to be reported.

Adjust the times if:

- The CPU clock of your OS/390 system is not set to the local time, but you want to use the local time in your reports.
 The CPU clock can be set to Greenwich Mean Time (GMT) or to the local time of another location, for example the local time of your head office.
- You want to generate reports or traces that show activity at more than one location and the CPU clock settings of the locations are different. This is often the case when the locations are in different time zones.

The data for calculating the adjusted times is stored in the LOCDATA member of the DPMPARMS data set. You can enter and edit the data using the Interactive Report Facility (IRF).

“Time Zone Data Editor”

Time zone information is entered with the Time Zone Data Editor, which is accessed through the Interactive Report Facility (IRF).

“Examples of time zone processing” on page 247

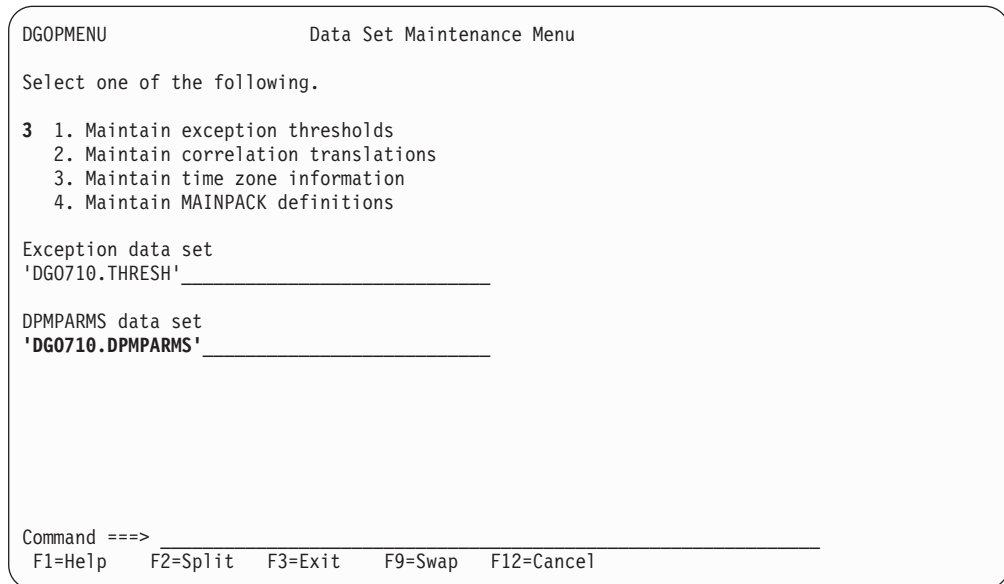
In the following examples, the settings in the Time Zone column show the real time zones relative to GMT. These settings can be maintained to reflect seasonal differences in daylight saving.

Time Zone Data Editor

Time zone information is entered with the Time Zone Data Editor, which is accessed through the Interactive Report Facility (IRF).

About this task

To access the editor, select option 4 (Maintain parameter data sets) from the Performance Expert main menu. The Data Set Maintenance Menu is displayed.



```
DGOPMENU                                Data Set Maintenance Menu

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
'DG0710.THRESH' _____

DPMPARMS data set
'DG0710.DPMPARMS' _____

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

Figure 56. Selecting Maintain Time Zone Information

Type 3 in the input field to select Maintain time zone information, and type the name of your DPMPARMS data set on the line below DPMPARMS data set.

Press Enter to go to the Time Zone Data Editor panel.

Use the editor to enter time zone specifications for the reporting locations. Specify the location, the geographical time zone, and the CPU clock setting of the z/OS system under which the DB2 subsystem runs.

DG0PPLDS

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 57. 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's Store Clock instruction value of the location and GMT.

You can get information about each entry field by moving the cursor to the field and pressing F1 (Help) to display the help text for the field.

Examples of time zone processing

In the following examples, the settings in the Time Zone column show the real time zones relative to GMT. These settings can be maintained to reflect seasonal differences in daylight saving.

The values in the CPU clock column are for illustration purposes only. You have 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” on page 248
This example shows how a specific location is assigned to a time zone and how this location's time is reflected in reports.

“Example 2: CPU clock settings” on page 249

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.

“Example 3: Time zones and CPU clock settings” on page 250

This example shows how to report data from two locations. The first location, USIBMSTOSQL1, is situated in San Francisco, the second, USIBMSTOSQL2, in New York.

Example 1: Time zones

This example shows how a specific location is assigned to a time zone and how this location's time is reflected in reports.

The figure shows the contents of the LOCDATA member of DPMPARMS. In this example, we want to report data from a DB2 subsystem in San Francisco. The location name of the DB2 subsystem is USIBMSTOSQL1.

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 58. Time Zone Data 1

The Time Zone column shows the time zone of the location relative to GMT. San Francisco is 8 hours west of Greenwich.

No value is specified in the CPU Clock column to indicate that the CPU clock is set to GMT. If you do not specify a value for the CPU clock, you do not have to change the default direction (1).

In this example a default (*) is specified for locations that do not have an individual entry. If you do not specify a default, no time zone adjustment is made to data from locations that do not have an entry.

If you want to use San Francisco local time in your reporting, specify location USIBMSTOSQL1 as your TIMEZONE location:

```
⋮
GLOBAL
  TIMEZONE (USIBMSTOSQL1)
⋮
```

The following adjustments are applied during OMEGAMON XE for DB2 PE processing:

- Data from location USIBMSTOSQL1 is adjusted -8 hours because its CPU clock is set to GMT and its local time is eight hours less than GMT.

- Data from locations that do not have an entry is adjusted -8 hours because the default CPU clock setting (*) is GMT as shown in Figure 58 on page 248.

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 59. 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 Performance Expert processing:

- Data from location USIBMSTOSQL1 is adjusted -3 hours because its CPU clock is set to New York local time, which is 5 hours less than GMT.
- Data from locations that do not have an entry is adjusted -8 hours because the default CPU clock setting (*) is GMT, as shown in Figure 59.

As a result, data from all locations reflects San Francisco local time.

Example 3: Time zones and CPU clock settings

This example shows how to report data from two locations. The first location, USIBMSTOSQL1, is situated in San Francisco, the second, USIBMSTOSQL2, in New York.

```

DGOPPLDS                               Time Zone Data Editor                               Row 1 to 3 of 3

Direction (1=East 2=West)
Action  Location                        --   Time Zone   --   --   CPU Clock   --
-----
      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 60. 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.

Again, if you want to use San Francisco local time in your reporting, specify location USIBMSTOSQL1 as your TIMEZONE location:

```
⋮
GLOBAL
  TIMEZONE (USIBMSTOSQL1)
⋮
```

The following adjustments are made during Performance Expert processing:

- Data from location USIBMSTOSQL1 is adjusted -8 hours because its CPU clock is set to GMT.
- Data from location USIBMSTOSQL2 is adjusted -3 hours because its CPU clock is set to New York local time, which is 5 hours less than GMT.
- Data from locations that do not have an entry is adjusted -8 hours because the default CPU clock setting (*) is GMT, as shown in Figure 60.

As a result, data from all locations reflects San Francisco local time.

For more information about time zone processing, refer to the description of the GLOBAL command in the *Report Command Reference*.

Correlation ID translation

The correlation ID is a DB2 field that identifies the task executed by DB2.

The correlation ID contains:

For batch jobs

Jobname

For TSO applications

Original authorization ID (the logon user ID)

For applications using the DB2 call attachment facility

Original authorization ID (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).

“The default translation”

OMEGAMON XE for DB2 PE breaks the correlation ID into parts by translating the correlation ID into two separate identifiers, the *correlation name* and the *correlation number*.

“Changing the default translation” on page 252

You can override this default translation using the IRF option for maintaining parameter data sets (option 4 on the DB2 PM main menu).

The default translation

OMEGAMON XE for DB2 PE breaks the correlation ID into parts by translating the correlation ID into two separate identifiers, the *correlation name* and the *correlation number*.

Unless it was changed in your installation, this translation is based on the connection type of the thread and is done as follows:

Table 19. The 12-Byte Correlation ID field and the default translation

Connection Type	1	2	3	4	5	6	7	8	9	10	11	12
Batch	Correlation name: job name								Correlation number: blank			
TSO, DB2 call attach	Correlation name: original authorization ID								Correlation number: blank			
CICS	Correlation number: pool thread				Correlation name: transaction ID							
IMS	Correlation number: application PSBNAME				Correlation name: application PST							
RRS	Correlation name: the first 8 characters of the correlation ID provided by the application during signon								Correlation number: the remaining 4 characters			

The correlation name and correlation number can be used as OMEGAMON XE for DB2 PE identifiers CORRNAME and CORRNMBR in all report sets.

You can use CORRNAME and CORRNMBR to order data on reports and to include and exclude input records.

Changing the default translation

You can override this default translation using the IRF option for maintaining parameter data sets (option 4 on the DB2 PM main menu).

The correlation translation information is kept in the CORRDATA member of the DPMPARMS data set.

Each record in the CORRDATA member specifies the translation that is to be used for a specific connection ID. The connection ID is used here, not the connection type.

The translation is expressed as:

- Offset where the correlation name starts
- Length of the correlation name
- Offset where the correlation number starts
- Length of the correlation number

If OMEGAMON XE for DB2 PE does not find the connection ID for a given thread in the CORRDATA member, the default translation is used.

To activate the tailored correlation translation, specify the DPMPARMS data set for the DPMPARMS ddname.

Defining the MAINPACK identifier

In the Accounting report set you can use the MAINPACK identifier to distinguish plans according to the packages they contain. The representative package is either the first or the last package or DBRM executed within a plan.

This identifier is useful when the name of a plan does not provide satisfactory identification, as is the case with DBATs initiated by non-DB2 requesters that all have the same plan name DISTSERV.

You can define certain aspects of the MAINPACK identifier:

- Whether the first or the last package executed within a plan is used as the MAINPACK.
- Whether you want to use the package ID, the collection ID, or the location name of the package name as the value for the identifier. In the case of a DBRM, the program name is always used.

If you wish, you can have different MAINPACK definitions for data from different environments and from different plans.

The MAINPACK identifier can be used to include, exclude, and order data.

When you include data using MAINPACK, data from other packages belonging to the same plan is also reported. (If you used the PACKAGE identifier instead, data for that package, regardless of the plan, would be reported.) The same applies to ordering data.

The default definition for MAINPACK is to use the package ID of the first executed package.

The MAINPACK definition is stored in the member MAINPACK of the DPMPARMS data set. You can access the member using the MAINPACK Definition Member Editor, which is part of the Interactive Report Facility (IRF).

To access the editor, select option 4 (Maintain parameter data sets) from the Performance Expert main menu. The Data Set Maintenance Menu is displayed.

```
DGOMENU                                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
'DG0710.THRESH'_____

DPMPARMS data set
'DG0710.DPMPARMS'_____

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

Figure 61. 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.

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 62. 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. DB2 uses this name to identify the requester DB2 subsystem for distributed threads. For nondistributed threads this is the same as the local location name.

Connection ID

The 8-byte connection name used by DB2 to identify your environment.

Plan Name

The 8-byte name of the DB2 plan. In most cases it is a user-specified name, but for non-DB2 requesters it has a constant value DISTSERV.

Code One of the predefined definitions of MAINPACK.

You can get information about an entry field by moving the cursor to the field and pressing F1 (Help).

“Specifying the MAINPACK (example)” on page 255

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.

“Examples of using the MAINPACK and PACKAGE identifiers” on page 256

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.

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 CICSА 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 CICSА is a large plan consisting of several packages and it is used for many different kinds of transactions. The goal is to distinguish between the different executions of this plan.

The following figure shows the MAINPACK Definition Member Editor panel with the specifications.

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
_____	*	*	CICSА	2

***** Bottom of data *****

Command ==> _____

F1=HelpF2=SplitF3=ExitF7=UpF8=DownF9=SwapF12=Cancel

Figure 63. 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 provides the necessary information to identify the plan. The assumption for this plan was that the package identifier was the most convenient identifier value.

For plan CICSА, the representative package was defined as the last executed package. The reason for doing this was that for this particular plan, in this example, the last executed package best identifies the transaction. The package ID was used as the value of the identifier.

Examples of using the MAINPACK and PACKAGE identifiers

The following examples show how you can use the MAINPACK and PACKAGE identifiers. The first example is a short report and shows the short format of the package information. The second example shows the long format of the package information. Both examples show how to use these identifiers in INCLUDE/EXCLUDE and ORDER processing.

Example 1: Using MAINPACK

Assume that, during the reporting interval used in this example, two transactions are performed on behalf of a non-DB2 requester. One transaction executes package CUSTINQR and the other transaction executes package ACCTCLOS. You only want to report the activities that are performed by these non-DB2 requesters, that is, DISTSERV plans.

If you use PLANNAME to order the report, you cannot distinguish between the two packages or the two transactions because they both execute plan DISTSERV. So to identify the transactions by means of a package, the MAINPACK identifier is used in the ORDER. For this case the default specification for MAINPACK is used, which is to select the package ID of the first package executed within a plan. The Accounting report is then summarized and ordered by this MAINPACK (first package ID).

The following command was used:

```
:  
:  
ACCOUNTING  
  REPORT  
    INCLUDE (PLANNAME(DISTSERV))  
    ORDER  (MAINPACK)  
:
```

Short Accounting report - ordered by MAINPACK

Here is an example of a short Accounting report that is ordered by MAINPACK.

Note: The values for the package in the Package Information block represent the processing that is specific to that package. In this case, there is only one package executed for each transaction. If there had been a second package within the same transaction, then the second package would have been listed as a second line within the Package Information block.

LOCATION: VTAMA		OMEGAMON XE for DB2 Performance Expert (V5R3M0)							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: 04/29/13 20:18:50.43						
DB2 VERSION: V11		SCOPE: MEMBER							TO: 05/20/13 17:57:56.76						
MAINPACK		#OCCURS	#ROLLBK	SELECTS	INSERTS	UPDATES	DELETES	CLASS1	EL.TIME	CLASS2	EL.TIME	GETPAGES	SYN.READ	LOCK	SUS
		#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		

Example 2: Using PACKAGE

Assume that, during the reporting interval used in this example, three transactions are performed and that DB2 Accounting Trace classes 7 and 8 are active.

- The first transaction executes plan PLANINQ1 and uses packages CUSTINQA and CUSTINQB and DBRM CUSTINQC.
- The second transaction executes plan PLANINQ2 and uses packages CUSTINQB and CUSTINQD.
- The third transaction executes plan PLANINQ1 also but uses package CUSTINQA only.

All the packages are executed once within the transaction and the transactions are executed only once.

If you want to find out the resource utilization for each package or DBRM, you ORDER the report by PACKAGE and print only the package or DBRM specific data from the Accounting records. In this example, you want detailed information about the packages or DBRM, therefore you request a LONG report.

You have previously determined that you are not interested in package CUSTINQD, so you exclude that package. By default all plan names present in the input data are included in the report.

You specify the following command:

```
⋮  
ACCOUNTING  
REPORT  
    LAYOUT (LONG)  
    EXCLUDE (PACKAGE(CUSTINQD))  
    ORDER (PACKAGE)  
⋮
```

Long Accounting report - ordered by PACKAGE

Here is an example of a long Accounting report that is ordered by PACKAGE. It shows the Package Information blocks for the different packages. In this example, each entry begins on a new page, even though all entries fit on a single page.

On this report:

- Information for package CUSTINQA is derived from the first and the third transaction.
- Information for package CUSTINQB is derived from the first and the second transaction.
- Information for DBRM CUSTINQC is derived from the first transaction.

To print all entries on one page you can create your own layout by modifying the sample layout LONG. You can do this without changing the actual layout. Simply press F3 in the UTR Block Selection panel to display the UTR Save Layout Details panel (see Figure 55 on page 244). 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.

LOCATION: VTAMA	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 1-1
GROUP: N/P	ACCOUNTING REPORT - LONG	REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P		TO: NOT SPECIFIED
SUBSYSTEM: SSDQ	ORDER: PACKAGE	INTERVAL FROM: 04/29/13 20:18:50.43
DB2 VERSION: V10	SCOPE: MEMBER	TO: 05/20/13 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

LOCATION: VTAMA	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	PAGE: 1-2
GROUP: N/P	ACCOUNTING REPORT - LONG	REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P		TO: NOT SPECIFIED
SUBSYSTEM: SSDQ	ORDER: PACKAGE	INTERVAL FROM: 04/29/13 20:18:50.43
DB2 VERSION: V10	SCOPE: MEMBER	TO: 05/20/13 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

LOCATION: VTAMA	OMEGAMON XE for DB2 Performance Expert (V5R3M0)	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: 04/29/13 20:18:50.43
DB2 VERSION: V10	SCOPE: MEMBER	TO: 05/20/13 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

Modifying exception field descriptors

You can change the descriptive labels for exception fields as they are printed for the exception records in batch reports.

To do this, create a member called EXCHANGE in your DPMPARMS data set using the ISPF editor. For each field that requires a new descriptor, enter a line into this member that contains the field name and the new descriptor, separated by a comma without spaces.

Chapter 24. The Performance Database and the Performance Warehouse

OMEGAMON XE for DB2 PE performance data can be loaded into DB2 tables to build a performance database or a performance warehouse.

You can load performance data into DB2 tables to create a performance database. With a performance database you can:

- Perform additional performance evaluations. As an example you can examine the access paths of dynamic SQL statements.
- Collect historic data.
- Use SQL for fast and easy retrieval of data.

The performance data can come from the following data groups:

- Accounting
- Audit
- Locking
- Record traces (IFCID 22, 63, 96, 125, 172, 196, 316, 365, and 401)
- Statistics
- System parameters
- Batch, periodic, and display exceptions

Data can be aggregated or nonaggregated:

- In the case of aggregated data, several records are summarized by specific OMEGAMON XE for DB2 PE identifiers. In a report, each entry represents aggregated data. Use the SAVE subcommand to generate a VSAM data set that contains the aggregated data. When the data has been saved, use the Save-File utility to generate a DB2-loadable data set.
- In the case of nonaggregated data, each record is listed in order of occurrence. In a trace, each entry represents nonaggregated data. Use the FILE subcommand to generate a data set that contains the nonaggregated data.

For batch, periodic, and display exceptions, a DB2-loadable data set is automatically generated for each OMEGAMON XE for DB2 PE execution if EXFILDD1 is defined in your JCL. The generated data set is sequential variable-blocked and can be directly loaded into DB2.

The following figure shows how performance data is formatted and loaded for each data group.

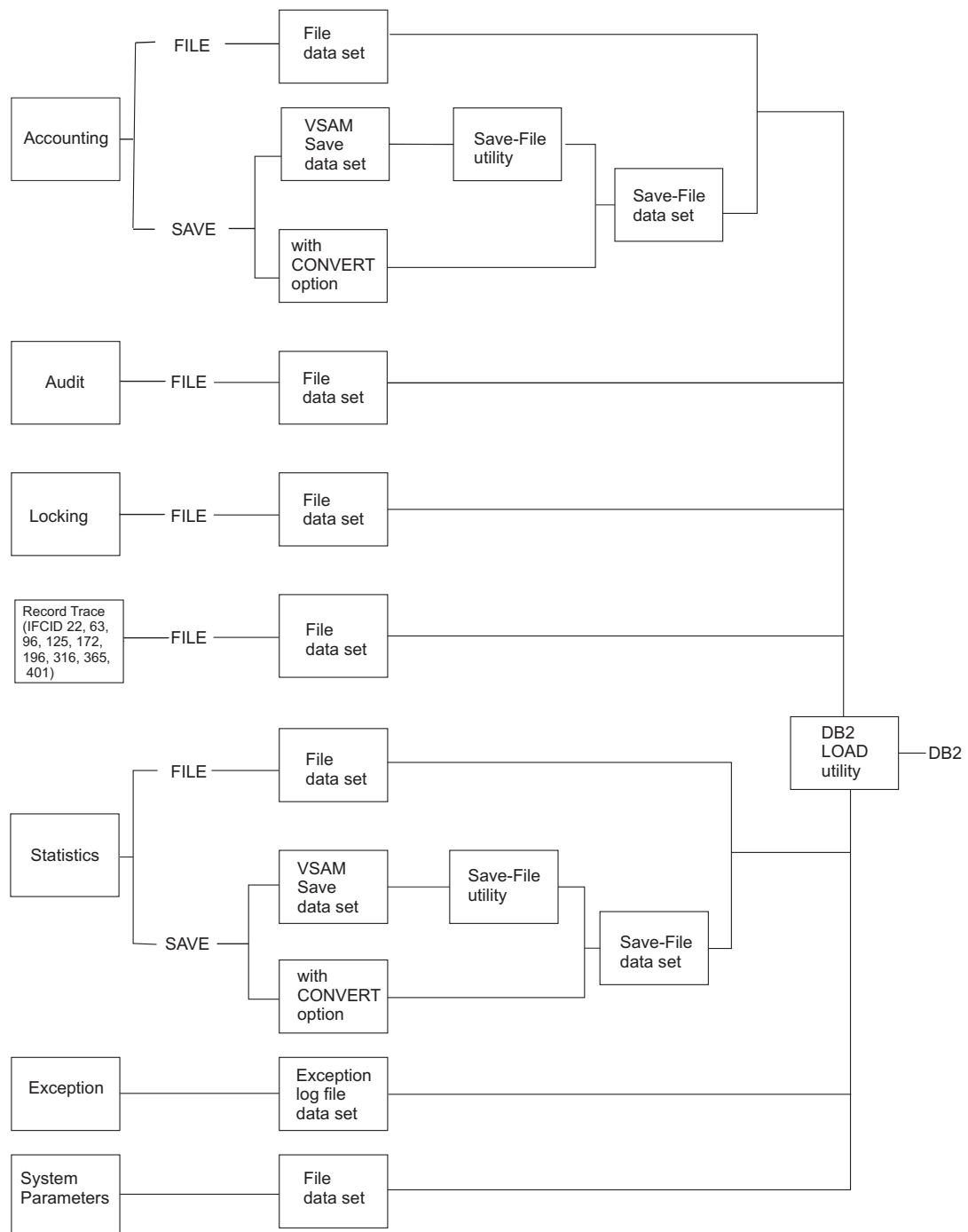


Figure 64. How performance data is formatted and loaded

For details about loading performance data into DB2 tables, consult the RKO2SAMP library.

“Accounting tables” on page 263

This section shows the structure of each of the Accounting tables in the performance database.

“Audit tables” on page 265

This section shows the structure of each of the Audit tables in the performance database.

“Exceptions table” on page 267

This section shows the structure of the Exceptions table in the performance database.

“Locking table” on page 267

This section shows the structure of the Locking table in the performance database.

“Record Trace tables” on page 268

This section shows the structure of each of the Record Trace tables in the performance database.

“Statistics tables” on page 270

This section shows the structure of each of the Statistics tables in the performance database.

“System parameters” on page 273

This section lists the system parameters tables in the performance database.

“Creating data for the Performance Database and the Performance Warehouse” on page 273

To create performance data, you must run the appropriate OMEGAMON XE for DB2 PE command with the FILE or SAVE subcommand.

Accounting tables

This section shows the structure of each of the Accounting tables in the performance database.

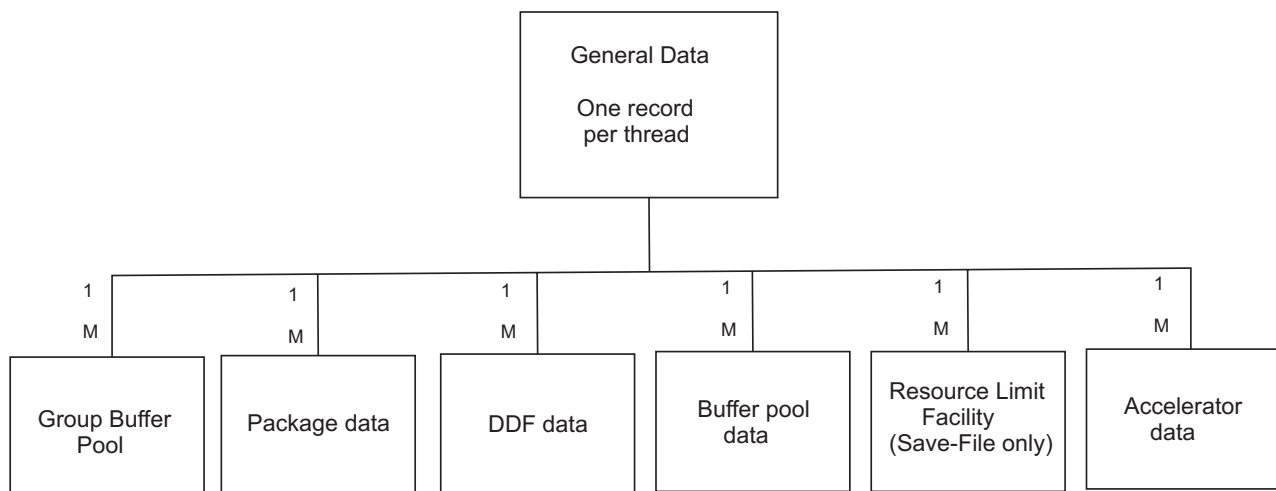


Figure 65. The Accounting tables

The generated data set consists of the following records:

General

One row per thread.

Group buffer pool

One row per group buffer pool used.

Package data

One row per package and DBRM executed.

DDF data

One row per remote location participating in distributed activity.

Buffer pool data

One row per buffer pool used.

RLF One row per resource limit type encountered.

Accelerator data

One row per thread-related accelerator activity.

When parallel tasks are performed within a data sharing group, the parallel task records are contained in the record of the originating task, regardless of whether the parallel tasks were performed on the same member as the originating task or on other members of the data sharing group.

“Samples”

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP.

The description files are in a format that can be loaded into DB2 tables.

Samples

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP.

The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE examples can be used for building the DB2 tables into which Accounting File and Save-File data can be loaded. The DB2 load control statements can be used for loading this data into the DB2 tables.

“Accounting File”

Accounting data can contain all or a subset of available data types, depending on the specification of the DATATYPE subcommand option of the ACCOUNTING FILE command.

“Accounting Save-File” on page 265

The parts for Accounting Save-File data are as follows.

Accounting File

Accounting data can contain all or a subset of available data types, depending on the specification of the DATATYPE subcommand option of the ACCOUNTING FILE command.

The specified data types can be stored in one or several data sets in any combination by means of multiple invocations of FILE DATATYPE in a JCL. To load all Accounting data from several output data sets into a DB2 table, you can run the following DB2 load control statements concurrently. For more information, see the *Report Command Reference*.

Table 20. Parts for Accounting file data

Type of data	CREATE TABLE statements	LOAD control statements	Table description
General	DGOACFGE	DGOALFGE	DGOABFGE
Group Buffer Pool	DGOACFGP	DGOALFGP	DGOABFGP
Buffer Pool	DGOACFBU	DGOALFBU	DGOABFBU
DDF Records	DGOACFDF	DGOALFDF	DGOABFDF
Package Records	DGOACFPK	DGOALFPK	DGOABFPK
Accelerator	DGOACFXC	DGOALFXC	DGOABFXC

Examples of SQL queries are in member DGOAQFIL.

Accounting Save-File

The parts for Accounting Save-File data are as follows.

Table 21. Parts for Accounting Save-File data

Type of data	CREATE TABLE statements	LOAD control statements	Table description
General	DGOACSGE	DGOALSGE	DGOABSGE
Group Buffer Pool	DGOACSGP	DGOALSGP	DGOABSGP
Buffer Pool	DGOACSBU	DGOALSBU	DGOABSBU
DDF Records	DGOACSDF	DGOALSDF	DGOABSDF
Package Records	DGOACSPK	DGOALSPK	DGOABSPK
Resource Limit Facility (RLF) Records	DGOACSRF	DGOALSRF	DGOABSRF
Accelerator	DGOACSXC	DGOALSXC	DGOABSXC

VSAM-Save data sets must be converted to the Save-File layout before they can be loaded into tables.

Examples of CREATE VIEW statements are supplied in the member DGOAVSAV.

Examples of SQL queries are in member DGOAQSAV.

DGOAVSAV contains the VIEW definition necessary to integrate the Accounting Save General table into the SAP DBA Cockpit (see SAP Note 1995032).

Audit tables

This section shows the structure of each of the Audit tables in the performance database.

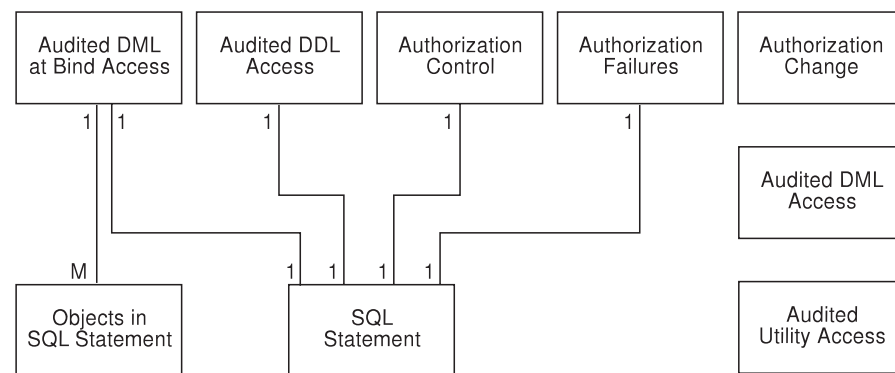


Figure 66. The Audit tables

Authorization Failures

One row for each authorization failure that occurred

Authorization Control

One row per authorization control issued

Authorization Change

One row for each of the following authorization changes or authorization establishments:

- SET CURRENT SQLID
- END OF IDENTIFY
- END OF SIGNON
- INBOUND/OUTBOUND DISTRIBUTED TRANSLATION

Audited DDL Access

One row per CREATE or DROP against an Audited object, or ALTER TABLE which change the AUDIT option against an audited table.

Audited DML at Bind Access

One row for each SQL statement that involves audited objects.

Objects in SQL statement

One row for each object involved in the SQL statements recorded in Audited DML at Bind Access.

Audited DML Access

One row for the first change (write) to an audited object in a unit of work

Audited Utility Access

One row for each object and each phase accessed by a utility execution.

SQL Statement

One row for the text of each SQL statement involved in:

- Authorization Failure
- Authorization Control
- Audited DDL Access
- Audited DML at Bind Access

“Samples”

The parts for Audit file data are as follows.

Samples

The parts for Audit file data are as follows.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE examples can be used for building the DB2 tables into which the Audit File data sets can be loaded. The DB2 load utility control statements can be used for loading these Audit File output records into DB2 tables.

Table 22. Parts for Audit file data

Type of data	CREATE TABLE statements	LOAD control statements	Table description
Bind	DGOXCBND	DGOXLBND	DGOXBBND
Bind Repeating Data	DGOXCBRD	DGOXLBRD	DGOXBBRD
Auth Change	DGOXCCHG	DGOXLCHG	DGOXBCHG
Auth Control	DGOXCCNT	DGOXLCNT	DGOXBCNT
DDL	DGOXCDDL	DGOXLDDL	DGOXBDDL
DML	DGOXCDML	DGOXLDML	DGOXBDML
Auth Fail	DGOXCFAI	DGOXLFAI	DGOXBFAI
SQL	DGOXCSQL	DGOXLSQL	DGOXBSQL

Table 22. Parts for Audit file data (continued)

Type of data	CREATE TABLE statements	LOAD control statements	Table description
Utility	DGOXCUTI	DGOXLUTI	DGOXBUTI

Exceptions table

This section shows the structure of the Exceptions table in the performance database.

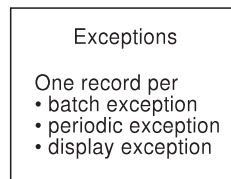


Figure 67. The Exception table

The generated table contains a list of Accounting and Statistics exceptions:

Batch exceptions

One row per exception

Periodic exceptions

One row per exception

Display exceptions

One row per exception

“Samples”

The parts for Exceptions file data are as follows.

Samples

The parts for Exceptions file data are as follows.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE example can be used for building the DB2 tables into which File data sets can be loaded. The DB2 load utility control statements can be used for loading these File output records into DB2 tables.

The query in DGOEQFIL uses the sample view defined in DGOEVFIL.

Table 23. Parts for Exceptions file data

Type of data	CREATE TABLE statements	LOAD control statements	Sample query	Table description
Exception	DGOECFIL	DGOELFIL	DGOEQFIL	DGOEBFIL

Locking table

This section shows the structure of the Locking table in the performance database.

Locking One record per • lock request • drain request • claim request • successful lock avoidance
--

Figure 68. The Locking table

The table contains a row for each occurrence of the following events:

- A LOCK, UNLOCK, CHANGE, or QUERY request processed by DB2.
- A request to acquire a claim, change a claim duration, or release a claim.
- A request to release a drain on a claim class.
- A successful lock avoidance.

“Samples”

The parts for Locking file data are as follows.

Samples

The parts for Locking file data are as follows.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE example can be used for building the DB2 tables into which the Locking File data sets can be loaded. The DB2 load utility control statements can be used for loading these Locking File output records into DB2 tables.

Table 24. Parts for Locking file data

Type of data	CREATE TABLE statements	LOAD control statements	Table description
Bind	DGOLCFIL	DGOLLFIL	DGOLBFIL

Record Trace tables

This section shows the structure of each of the Record Trace tables in the performance database.

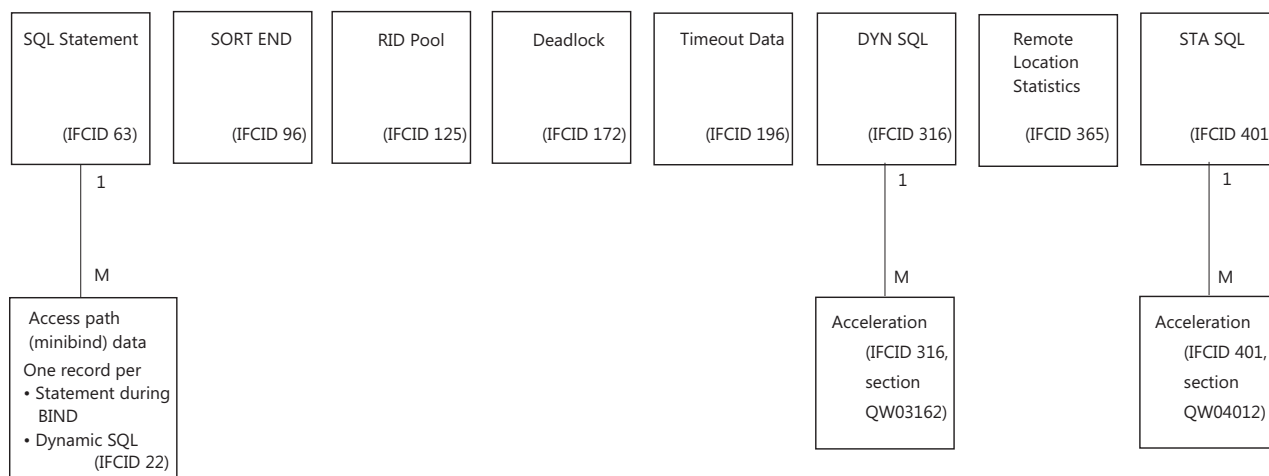


Figure 69. The Record Trace data set

The generated data set contains the following records:

Access path

Minibind data. One row per IFCID 22.

SQL Statement

One row per SQL statement text record (IFCID 63).

SORT END

One row for each sort end event triggered by an SQL query (IFCID 96).

RID Pool

One row per multiple index access operation (IFCID 125).

Deadlock (Unit of Work)

One row per resource involved (IFCID 172).

Timeout Data

One row per agent causing the timeout (IFCID 196).

SQL Statement Statistics (DYN SQL)

One row per SQL statement statistics text record (IFCID 316).

Acceleration

One row per accelerator to which the statement has been offloaded (IFCID 316, section QW03162).

Remote Location Statistics

One row per remote location (IFCID 365).

Static Statements in EDM Pool (STA SQL)

One row per static statements text record in the EDM pool (IFCID 401).

Acceleration

One row per accelerator to which the statement has been offloaded (IFCID 401, section QW04012).

“Samples”

The parts for Record Trace File data are as follows.

Samples

The parts for Record Trace File data are as follows.

Examples of CREATE TABLE and LOAD statements are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE examples can be used for building the DB2 tables into which Record Trace File data sets can be loaded. The DB2 load utility control statements can be used for loading these Record Trace File output records into DB2 tables.

Table 25. Parts for Record Trace File data

Type of data	CREATE TABLE statements	LOAD control statements	Sample query	Table description
Minibind (IFCID 22)	DGONCFMB	DGONLFMB	DGONQFMB	DGONBFMB
SQL Statement (IFCID 63)	DGONCFSQ	DGONLFSQ	DGONQFSQ	DGONBFSQ
SORT END (IFCID 96)	DGONCFSE	DGONLFSE	DGONQFSE	DGONBFSE
RID POOL (IFCID 125)	DGONCFRP	DGONLFRP	DGONQFRP	DGONBFRP
Deadlock (IFCID 172)	DGONCFDL	DGONLFDL	DGONQFDL	DGONBFDL
Timeout Data (IFCID 196)	DGONCF TI	DGONLFTI	DGONQFTI	DGONBFTI
SQL Statement Statistics (DYN SQL) (IFCID 316)	DGONCFDS	DGONLFDS	DGONQFDS	DGONBFDS
Dynamic Statement Acceleration (IFCID 316, section QW03162)	DGONCFD2	DGONLFD2		DGONBFD2
Remote Location Statistics (IFCID 365)	DGONCFRL	DGONLFRL	DGONQFRL	DGONBFRL
Static Statements in EDM Pool (STA SQL) (IFCID 401)	DGONCFSS	DGONLFSS	DGONQFSS	DGONBFSS
Static Statement Acceleration (IFCID 401, section QW04012)	DGONCFS2	DGONLFS2		DGONBFS2

Statistics tables

This section shows the structure of each of the Statistics tables in the performance database.

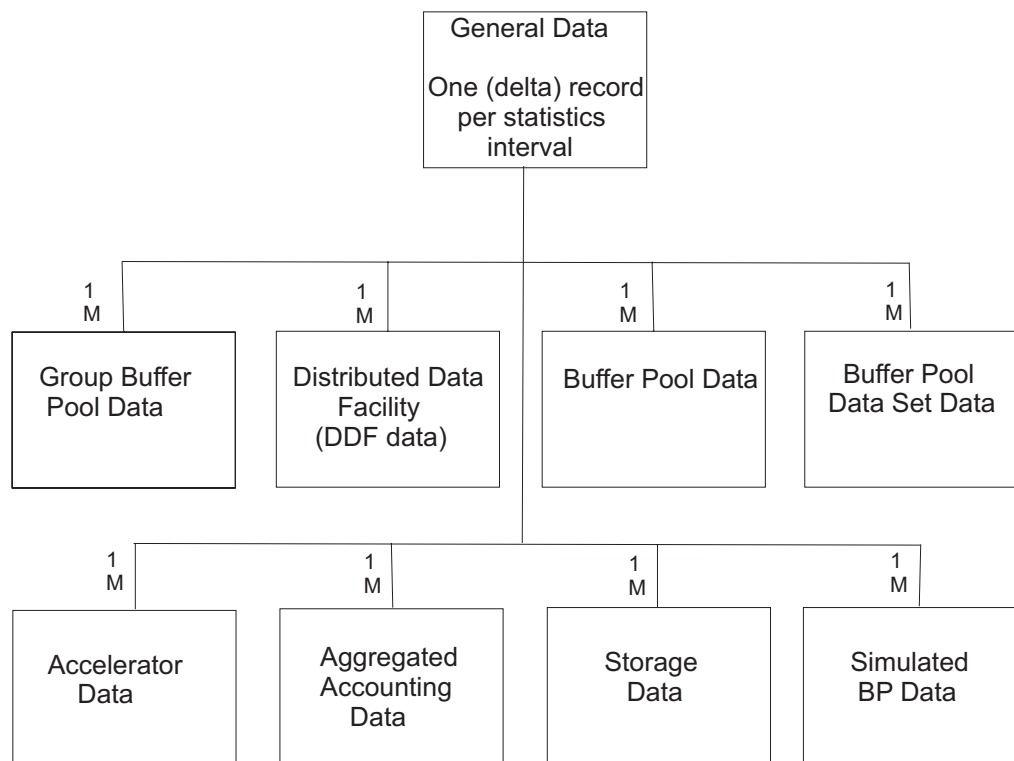


Figure 70. The Statistics table

The generated table contains the following records:

General Data

File Data

One row for each Statistics delta record, containing data from IFCID 1 and 2.

A delta record is a set of counters that describes the DB2 activity between two consecutive DB2 Statistics record pairs.

Save-File Data

One row for each Statistics interval record, containing data from IFCID 1 and 2. A Statistics interval record is a set of counters that describes the DB2 activity within the interval specified by the user.

Buffer Pool Data

One row per buffer pool active at the start of the corresponding delta record for File data, or interval record for Save-File data (containing data from the QBST section of IFCID 2).

Group Buffer Pool Data

One row per group buffer pool active at the start of the corresponding delta record for File data, or interval record for Save-File data (containing data from the QBGL section of IFCID 2).

DDF For each delta record (File data) or interval record (Save-File data), one row for all remote locations that used DRDA (containing data from the QLST section of IFCID 1).

Buffer Pool Data Set Data

One row for each open data set that has an I/O event rate at least one event per second during the reporting interval (containing data from IFCID 199).

Accelerator Data

One row per active accelerator attached to the DB2 subsystem (containing data from the Q8ST section of IFCID 2).

Aggregated Accounting Data

One row is written for a connection type IMS, CICS, RRSAP, Utility, Batch, or DDF containing aggregated wait and CPU times of threads with IFCID 3 events for that connection type (containing data from IFCID 369).

Storage Data

One row is written for DB2 storage metrics valid for the reporting interval (containing data from IFCID 225).

Simulated Buffer Pool Data

One row per simulated buffer pool active at the start of the corresponding delta record for File data, or interval record for Save-File data (containing data from the QBSP section of IFCID 2).

“Samples”

The parts for Statistics File and Save-File data are as follows.

Samples

The parts for Statistics File and Save-File data are as follows.

Examples of CREATE TABLE, LOAD statements, SQL queries, and descriptions of the statistics tables are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE examples can be used for building the DB2 tables into which the Statistics File data sets and the converted Statistics Save records can be loaded. The DB2 load utility control statements can be used for loading these Statistics File data sets and converted Statistics Save records into DB2 tables.

Table 26. Parts for Statistics File and Save-File data

Type of data	CREATE TABLE statements	LOAD control statements	Sample query	Table description
General Data	DGOSCGEN	DGOSLGEN	DGOSQGEN	DGOSBGEN
Buffer Pool	DGOSCBUF	DGOSLBUF	DGOSQBUF	DGOSBBUF
DDF	DGOSCDDF	DGOSLDDF	DGOSQDDF	DGOSBDDF
Group Buffer Pool	DGOSCGBP	DGOSLGBP		DGOSBGBP
Buffer Pool Data Set	DGOSCSET	DGOSLSET		DGOSBSET
Accelerator	DGOSCXCL	DGOSLXCL		DGOSBXCL
Aggregated Accounting	DGOSCACC	DGOSLACC		DGOSBACC
Storage Data	DGOSCSTG	DGOSLSTG		DGOSBSTG
Simulated BP Data	DGOSCSIM	DGOSLSIM		DGOSBSIM

The sample query in DGOSQDDF uses the view defined in DGOSVDDF.

The samples provided are valid for Statistics File and Save-File data.

Save data sets must be converted to the Save-File layout before they can be loaded into tables.

System parameters

This section lists the system parameters tables in the performance database.

The Performance Database tables for DB2 system parameters are as follows:

- System parameters from IFCID 106
- System facility parameters from IFCID 106
- Alter buffer pool records from IFCID 201
- Buffer pool attributes from IFCID 202
- Group buffer pool attributes from IFCID 230
- Alter group buffer pool records from IFCID 256

To create system parameter data for the Performance Database, use the SYSPARMS command with the FILE subcommand to produce a data set that is suitable for loading into DB2 tables.

“Samples”

The parts for System Parameters data are as follows.

Samples

The parts for System Parameters data are as follows.

Examples of CREATE TABLE, LOAD statements, SQL queries, and descriptions of these tables are in library RKO2SAMP. The description files are in a format that can be loaded into DB2 tables.

The CREATE TABLE examples can be used for building the DB2 tables into which the System Parameters File data sets can be loaded. The DB2 load utility control samples can be used for loading these File data sets into DB2 tables.

Table 27. Parts for System Parameters data

Type of data	CREATE TABLE statements	Load control statements	Table description
System parameters	DGOWC106	DGOWL106	DGOWD106
System facility parameters	DGOWCSFP	DGOWLSFP	DGOWDSFP
Alter buffer pool	DGOWC201	DGOWL201	DGOWD201
Buffer pool attributes	DGOWC202	DGOWL202	DGOWD202
Group buffer pool attributes	DGOWC230	DGOWL230	DGOWD230
Alter group buffer pool	DGOWC256	DGOWL256	DGOWD256

Creating data for the Performance Database and the Performance Warehouse

To create performance data, you must run the appropriate OMEGAMON XE for DB2 PE command with the FILE or SAVE subcommand.

If you use the SAVE subcommand:

- Without the CONVERT option, you must first convert the VSAM data to the Save-File format.
- With the CONVERT option, the reduced data is converted and saved into a sequential data set.

You can then use the DB2 LOAD utility to move the data into DB2 Performance Database tables.

If you have installed the DB2 Performance Warehouse Client together with the corresponding host data collector, the following tasks can be simplified and largely automated for Accounting and Statistics data through the use of client-initiated processes:

- Creating performance data
- Loading the data into the Performance Database (Warehouse)
- Maintaining the database

For more information about the:

- Performance Warehouse, see *Monitoring Performance from Performance Expert Client*
- FILE or SAVE option, see “Functions and utilities of the Statistics report set” on page 193 and *Report Command Reference*
- Save-File Utility, see “Functions and utilities of the Statistics report set” on page 193 and *Report Reference*
- Spreadsheet Input-Data Generator utility, see Chapter 25, “Generating input data for spreadsheets,” on page 275

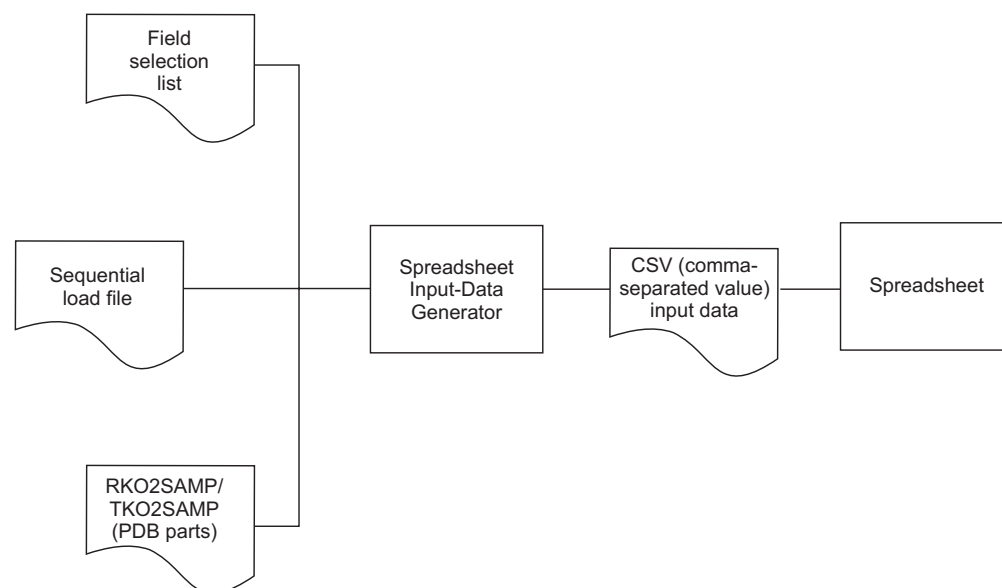
Chapter 25. Generating input data for spreadsheets

With the *Spreadsheet Input-Data Generator* utility you can create input data for spreadsheets to ease DB2 performance analysis.

This utility uses the following data as input:

- Accounting and Statistics File performance data
- Accounting and Statistics converted Save-File performance data
- Record Trace File data of IFCID 172, 196, and 365

It creates a *comma-separated value* (CSV) data set using input provided by parts of the Performance Database (PDB) in the **RKO2SAMP** / **TKO2SAMP** library and also your input from field selection lists. This CSV data can then be transferred to workstations and imported into spreadsheets to improve DB2 performance analysis using graphical representations or pivot tables.



Input parameters

The Spreadsheet Input-Data Generator mainly relies on the sample CREATE, LOAD, and meta data parts delivered for the Performance Database in the **RKO2SAMP** / **TKO2SAMP** library. These parts describe the record layout of the sequential load file input data, the positions of fields within the record, data type, length, WHEN, NULLIF, and DEFAULTIF values. Based on this information and the following input parameters, the fields of the sequential load file can be transformed into a CSV record.

The load module **FPEPCSV** is linked in library **RKANMOD** / **TKANMOD**. The following samples are provided in the **RKANSAM** / **TKANSAM** library:

FPEPCSVJ

The sample JCL to invoke the Spreadsheet Input-Data Generator utility.

FPEPCSV

Invokes the Spreadsheet Input-Data Generator utility with a job procedure. The following sample jobs are provided:

FPEPCSV

Invokes the Spreadsheet Input-Data Generator utility and creates multiple Accounting Save spreadsheets with a job procedure.

FPEPCSVB

Invokes the Spreadsheet Input-Data Generator utility and creates multiple Accounting File spreadsheets with a job procedure.

FPEPCSVN

Invokes the Spreadsheet Input-Data Generator utility and creates multiple Record Trace spreadsheets with a job procedure.

FPEPCSVS

Invokes the Spreadsheet Input-Data Generator utility and creates multiple Statistics spreadsheets with a job procedure.

FPEPCSVW

Invokes the Spreadsheet Input-Data Generator utility and creates multiple System Parameter spreadsheets with a job procedure.

The input parameters are specified in the EXEC command of the Spreadsheet Input-Data Generator in the parameter list **PARM** of **FPEPCSVJ**:

```
...
/* parm = type persec hdrrow hrdpdc col delim decsep sampdir
//FPEPCSV EXEC PGM=FPEPCSV,
// PARM='SBUF N Y N , . <db2ompe>.TKO2SAMP'
...
```

Where "parm = type persec hdrrow hrdpdc col delim decsep sampdir" specifies the following:

TYPE

Specifies the type of records that are put into the CSV data set. For each supported TYPE, a sample *master selection list* is provided that contains all fields in the sequential load file records that have a column in the associated Performance database (PDB) table.

The **RKO2SAMP** / **TKO2SAMP** part type consists of 4 characters, it can be:

AFBU Accounting File data, category BUFFER POOL

AFDF Accounting File data, category DDF

AFGE Accounting File data, category GENERAL

AFGP Accounting File data, category GROUP BUFFER POOL

AFPK Accounting File data, category PACKAGE

AFXC Accounting File data, category ACCELERATOR

ASBU Accounting Save-File data, category BUFFER POOL

ASDF Accounting Save-File data, category DDF

ASGE Accounting Save-File data, category GENERAL

ASGP Accounting Save-File data, category GROUP BUFFER POOL

ASPK Accounting Save-File data, category PACKAGE

ASRF Accounting Save-File data, category RLF

	ASXC	Accounting Save-File data, category ACCELERATOR
	NFDL	Record trace data, category DEADLOCK
	NFTI	Record trace data, category TIMEOUT
	NFRL	Record trace data, category REMOTE LOCATION
	SACC	Statistics data, category AGGREGATED ACCOUNTING
	SBUF	Statistics data, category BUFFER POOL
	SDDF	Statistics data, category DDF
	SGBP	Statistics data, category GROUP BUFFER POOL
	SGEN	Statistics data, category GENERAL
	SSET	Statistics data, category BUFFER POOL DATA SET
	SSIM	Statistics data, category BUFFER POOL DATA SET
	SSTG	Statistics data, category SIMULATED BUFFER POOL
	SXCL	Statistics data, category ACCELERATOR
	W106	System Parameter data, category IFCID 106
	W201	System Parameter data, category IFCID 201
	W202	System Parameter data, category IFCID 202
	W230	System Parameter data, category IFCID 230
	W256	System Parameter data, category IFCID 256
	WSFP	System Parameter data, category DDF Facility

PERSEC

Specifies whether average values are calculated. It can be Y (for YES) or N (for NO). This option is only valid for Statistics records where the interval column INTERVAL_ELAPSED can be used to derive per-second (PERSEC) values.

Note: Averages are only calculated for Statistics fields which are accumulated by DB2 and have a field TYPE of AGGR defined in a Statistics field selection list (FPEPS*).

HDRROW

Specifies whether a header row is added to the CSV output. It can be:

F (for FORCED)

Indicates that the header row is written in any case.

Y (for YES)

Indicates that the header row is written only if input data is available.

N (for NO)

Indicates that the header row is not written.

HDRPDCOL

Specifies whether the Performance Database (PDB) column name or the report label in the field selection list is used as header row. It can be Y (for YES) or N (for NO).

It is ignored if HDRROW=N.

If HDRPDCOL=N, the REPORT LABEL (V) entries are chosen as CSV column names. You can modify these entries according to your requirements.

DELIM

Specifies the CSV delimiter to be used in the output. It can be a comma (,) or a semicolon (;).

DECSEP

Specifies which decimal separator is used in the output. It can be a dot (.) or a comma (,):

Dot (.)

If DELIM is a comma (,), DECSEP must be a dot (.) and the CSV output is suited for workstation settings with dot as decimal separator.

Comma (,)

If DELIM is a semicolon (;), DECSEP must be a comma (,) and the CSV output is suited for workstation settings with comma as decimal separator.

SAMPDIR

Specifies the sample **RKO2SAMP** / **TKO2SAMP** library to be used.

The following DD names must be specified in **FPEPCSVJ**:

```
...
//STEPLIB DD DSN=<db2ompe>.TKANMOD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//FLDSEL DD DSN=<user_hlq>.FPECSVU(FPEPSBUF),DISP=OLD
//INPUT DD DSN=<user_hlq>.STFILE,DISP=OLD
//OUTPUT DD DSN=<user_hlq>.SCSVSBUF,DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,SPACE=<CYL,<150,150>,RLSE),
// DCB=<RECFM=VB,LRECL=32756,BLKSIZE=32760>
...
```

Where:

FLDSEL

Specifies the field selection lists you want to use.

For each supported **TYPE**, a sample *master selection list* is provided in the library that contains all fields in the sequential load file records that have a column in the associated Performance database (PDB) table.

The master field selection lists for all supported data sets of Accounting and Statistics File, converted Save-File, and Record Trace File are provided in the **RKANSAMF** / **TKANSAMF** library. The name of a field list *FPEPtype* consists of the component code FPEP and the record *type* described in input parameter **TYPE**. For more information refer to “Input parameters” on page 275. For example, FPEPSFBU is the sample field selection list for the Spreadsheet Input-Data Generator utility and Statistics File data, category BUFFER POOL.

INPUT

Specifies the data set of Accounting or Statistics File, converted Save-File, or Record Trace File.

OUTPUT

Specifies the CSV data set.

Input and output data sets

The input and output data sets have following characteristics:

- A field selection list (such as FPEPSGEN) specifies which fields from the sequential load file input data set are transformed to the CSV output.

- For each supported TYPE, a sample *master field selection list* is provided that contains all fields in the sequential load file records that have a column in the associated Performance database (PDB) table.

You can create subsets of these field selection lists for tailoring the CSV output to your needs. The record layout is as follows:

1. The first 13 rows contain the copyright statement. Comments must start with a forward slash followed by an asterisk (/*) and end with an asterisk followed by a slash (*). You can delete or update these comments.
2. The next two header rows and the first three columns are fixed (=F). You must not change them.
3. The fourth and fifth columns are variable (=V). You can modify them.

Here is a sample for the field selection list for Statistics data, category GENERAL (FPEPSGEN):

```

/****Start of Specifications*****/
/*
/* MODULE-NAME      = FPEPSGEN
/* DESCRIPTIVE-NAME = Field selection list for Spreadsheet Utility
/*
/* COPYRIGHT : IBM OMEGAMON XE for DB2 PE on z/OS V5
/*             Licensed Material - Property of IBM
/*             5655-W37 (C) Copyright IBM Corp. 1993, 2015
/*
/* FUNCTION = Sample field selection list for Spreadsheet Utility
/*            and Statistics data, category GENERAL.
/*
/****End of Specifications*****/
FIELD(F)  PDB COLUMN NAME(F)  TYPE (F)  FORMAT(V)  REPORT LABEL (V)
-----
DB2PMREL  DB2PM_REL           SNAP      DB2PM_REL
DB2REL    DB2_REL             SNAP      DB2_REL
QWHSSID   MVS_ID              SNAP      MVS_ID
QWHSLOCN  LOCAL_LOCATION      SNAP      LOCAL_LOCATION
QWHADSGN  GROUP_NAME          SNAP      GROUP_NAME
QWHSSSID  SUBSYSTEM_ID        SNAP      SUBSYSTEM_ID
QWHAMEMN  MEMBER_NAME         SNAP      DATA SHARING MEMBER
SDBEGINT  INTERVAL_TSTAMP     SNAP      INTERVAL_TSTAMP
SDELTIME  INTERVAL_ELAPSED    AGGR      INTERVAL_ELAPSED
SDBEGREC  BEGIN_REC_TSTAMP    SNAP      INTERVAL START
SDENDREC  END_REC_TSTAMP      SNAP      INTERVAL END
QXSELECT  SELECT              AGGR      SELECT
QXINSRT   INSERT              AGGR      INSERT
QXUPDTE   UPDATE              AGGR      UPDATE
QXMERGE   MERGE               AGGR      MERGE
QXDELET   DELETE              AGGR      DELETE
QXPREP    PREPARE             AGGR      PREPARE
QXDESC    DESCRIBE            AGGR      DESCRIBE
QXDSCRTB  DESCRIBE_TABLE      AGGR      DESCRIBE TABLE
QXOPEN    OPEN                AGGR      OPEN
QXFETCH   FETCH               AGGR      FETCH
QXCLOSE   CLOSE               AGGR      CLOSE
...

```

The record layout of a field selection list contains the following headers and columns:

FIELD (F)

The FIELD header and its column is fixed (=F). You must not change it.

PDB COLUMN NAME (F)

The PDB COLUMN NAME (Performance Database) header and its column is fixed (=F). You must not change it.

TYPE (F)

The TYPE header and its column is fixed (=F). You must not change it. The following values are possible:

AGGR

Aggregated value by DB2

HWM High-water mark

SNAP Snapshot value

FORMAT (V)

The FORMAT header and its column is variable (=V). You can modify it.

This column specifies special formatting routines. The following formats are supported:

MB Divides input value (bytes expected) by 1024*1024.

If MB is specified in the master field selection list, then (MB) is shown as a suffix of the report label.

P2MB Formats number of pages (4K-pages expected) as MB value.

If P2MB is specified in the master field selection list, then (MB) is shown as a suffix of the report label.

HEX Shows the hexadecimal value.

S2T Transforms a store-clock value to a readable time format.

BP2C Transforms buffer pool identifiers (small integers) in the BP_ID columns of Accounting and Statistics spreadsheets to a character representation, such as:

0 → BP0
100 → BP8K0

GBP2C

Transforms group buffer pool identifiers (small integers) in the GBP_ID columns of Accounting and Statistics spreadsheets to a character representation, such as:

0 → GBP0
100 → GBP8K0

REPORT LABEL (V)

The REPORT LABEL header and its column is variable (=V). You can modify it.

Note:

- If HDRPDCOL=N, the REPORT LABEL entries are chosen as CSV column names. You can modify these entries according to your requirements.
- The sample field selection lists contain report labels which are derived from the labels in Accounting and Statistics reports. Some report labels refer to the Performance database (PDB) column name.

Field selection lists are delivered in the **RKANSAMF** / **TKANSAMF** library with the following attributes:

Organization . . . : P0
Record format . . . : FB
Record length . . . : 132

The Spreadsheet Input-Data Generator expects field selection lists under DD name FLDSEL in a partitioned data set with the above attributes of the **RKANSAMF** / **TKANSAMF** library. This means that you can use the master field selection lists of **RKANSAMF** / **TKANSAMF** as input if you do not want to customize the lists. Otherwise, you should copy the sample lists to a partitioned data set with the above FB132 attributes and tailor them to your needs.

The *sequential load file* denotes the File or converted Save-File data set that is input for the Spreadsheet Input-Data Generator. The records of the input data set which satisfies the WHEN condition of the TYPE-related LOAD statement are read, the fields

specified in the selection list are retrieved from this input record, converted, and added to the CSV record according to the PERSEC and format options.

The Spreadsheet Input-Data Generator expects the input data set under DD name INPUT.

The **TKO2SAMP** directory is delivered with OMEGAMON XE for DB2 PE. It contains the PDB parts which are needed for the CSV creation (fields, positions, data types, and so on). The Spreadsheet Input-Data Generator expects the original product library as input. A run-time environment contains the parts in directory **RKO2SAMP**. The Spreadsheet Input-Data Generator expects the directory name as the 7-th input parameter after the decimal separator.

The output CSV data set must be provided to the Spreadsheet Input-Data Generator under DD name OUTPUT. The data set attributes are:

```
Organization . . . : PS
Record format . . . : VB
Record length . . . : 32756
Block size . . . . : 32760
```

Note: All input data sets must have the same OMEGAMON XE for DB2 PE version and maintenance level. The Spreadsheet Input-Data Generator:

- Checks the syntactical correctness of **RKO2SAMP** / **TKO2SAMP** members. If you use the original product library, the syntactical correctness is ensured. The Spreadsheet Input-Data Generator detects if members are changed or if they use the wrong syntax (such as unknown SQL syntax).
- Supports 1000 columns in one CSV row. For example, the Statistics File/Save General records currently contain about 600 fields.

Sample procedures and JCL

The Spreadsheet Input-Data Generator is invoked using PROCs and JCLs as shown in the following samples delivered with the **RKANSAM** / **TKANSAM** library. The following example creates a spreadsheet for Statistics data, category BUFFER POOL (SBUF) based on the master selection list <user_h1q>.FPECSVU(FPEPSBUF).

Use FPEPCSVJ to invoke the Spreadsheet Input-Data Generator utility.

```
/*Start of Specifications*****
/*
/*  MODULE NAME : FPEPCSVJ
/*
/*  DESCRIPTION : Spreadsheet Utility sample job
/*
/*  COPYRIGHT   : IBM OMEGAMON XE for DB2 PE on z/OS V5
/*                Licensed Material - Property of IBM
/*                5655-W37 (C) Copyright IBM Corp. 1993, 2015
/*
/*  FUNCTION    : Invoke spreadsheet utility
/*
/*  Notes =
/*    1. Add a valid job card
/*    2. Change the prefix of the OMEGAMON XE for DB2 PE
/*       library <db2ompe>
/*    3. Change the prefix of the user libraries <user_h1q>
/*    4. DDnames represent the following files...
/*      FLDSEL   User-modified field selection list
/*      INPUT    Accounting / statistics FILE or
/*              converted SAVE-FILE data set
/*      OUTPUT   Spreadsheet CSV data set
/*
/*      Verify the DD parameters (allocation, quantities, etc.)
/*
/*
/*End of Specifications*****
/* * parm = type persec hdrrow hdrpdc01 delim decsep sampdir
//FPEPCSV EXEC PGM=FPEPCSV,
//  PARM='SBUF N Y N , . <db2ompe>.TKO2SAMP'
//STEPLIB DD DSN=<db2ompe>.TKANMOD,DISP=SHR
```

```
//SYSPRINT DD SYSOUT=*
//FLDSEL DD DSN=<user_hlq>.FPECSVU(FPEPSBUF),DISP=OLD
//INPUT DD DSN=<user_hlq>.STFILE,DISP=OLD
//OUTPUT DD DSN=<user_hlq>.SCSVSBUF,DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(150,150),RLSE),
// DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
```

Multiple invocations of the OMEGAMON XE for DB2 PE Spreadsheet Input-Data Generator can be organized using job procedures with additional parameters for the input or output data sets. The &OUTDISP parameter specifies whether an existing CSV data set is overwritten or whether CSV records are appended to the end.

Use FPEPCSVU to invoke the Spreadsheet Input-Data Generator utility with a job procedure.

```
//*Start of Specifications*****
/*
/* MODULE NAME : FPEPCSVU
/*
/* DESCRIPTION : Spreadsheet Utility sample job procedure
/*
/* COPYRIGHT : IBM OMEGAMON XE for DB2 PE on z/OS V5
/* Licensed Material - Property of IBM
/* 5655-W37 (C) Copyright IBM Corp. 1993, 2015
/*
/* FUNCTION : Invoke spreadsheet utility with job procedure
/*
/* Notes =
/* 1. Change the prefix of the OMEGAMON XE for DB2 PE load
/* library <db2ompe>
/* 2. DDnames represent the following files...
/* FLDSEL User-modified field selection list
/* INPUT Accounting / statistics FILE or
/* converted SAVE-FILE data set
/* OUTPUT Spreadsheet CSV data set
/*
/* Verify the DD parameters (allocation, quantities, etc.)
/*
/*End of Specifications*****
//CSVPROC PROC TYPE=DUMMY,PERSEC=DUMMY,HDRROW=DUMMY,HDRPCOL=DUMMY,
// DELIM=DUMMY,DECSEP=DUMMY,SAMPDIR=DUMMY,
// FLDSSELS=DUMMY,INPUTDS=DUMMY,OUTPUTDS=DUMMY,OUTDISP=DUMMY
// EXEC PGM=FPEPCSV,
// PARM='&TYPE &PERSEC &HDRROW &HDRPCOL &DELIM &DECSEP &SAMPDIR'
//STEPLIB DD DSN=<db2ompe>.TKANMOD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//FLDSEL DD DSN=&FLDSSELS,DISP=OLD
//INPUT DD DSN=&INPUTDS,DISP=OLD
//OUTPUT DD DSN=&OUTPUTDS,DISP=&OUTDISP
// PEND
```

Use FPEPCSVS to invoke the Spreadsheet Input-Data Generator utility and create multiple Statistics spreadsheets with a job procedure.

```
//*Start of Specifications*****
/*
/* MODULE NAME : FPEPCSVS
/*
/* DESCRIPTION : Spreadsheet Utility sample job for creation of
/* STATISTICS FILE and SAVE spreadsheets
/*
/* COPYRIGHT : IBM OMEGAMON XE for DB2 PE on z/OS V5
/* Licensed Material - Property of IBM
/* 5655-W37 (C) Copyright IBM Corp. 1993, 2015
/*
/* FUNCTION : Invoke spreadsheet utility and create multiple
/* STATISTICS spreadsheets with a job procedure
/*
/* Notes =
/* 1. Add a valid job card
/* 2. Change the prefix of the OMEGAMON XE for DB2 PE
/* library <db2ompe>
/* 3. Change the prefix of the user libraries <user_hlq>
/* 4. DDnames represent the following files...
/* LIBS Library of job procedure FPEPCSVU
/*
/*End of Specifications*****
//-----
//LIBS JCLLIB ORDER=(<user_hlq>.FPEJCL)
//-----
/* STATISTICS -----
// EXEC PROC=FPEPCSV,TYPE=SACC,PERSEC=N,HDRROW=Y,HDRPCOL=N,
// DELIM=',',DECSEP='.',
// SAMPDIR=<db2ompe>.TKO2SAMP,
// FLDSSELS=<user_hlq>.FPECSVU(FPEPSACC),
```

```

//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSACC,
//      OUTDISP=OLD
// EXEC PROC=FPEPCSV,TYPE=SBUF,PERSEC=N,HDRROW=Y,HDRPCOL=N,
//      DELIM=' ',DECSEP='.',
//      SAMPDIR=<db2ompe>.TK02SAMP,
//      FLDSSELS=<user_h1q>.FPECSVU(FPEPSBUF),
//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSBUF,
//      OUTDISP=OLD
// EXEC PROC=FPEPCSV,TYPE=SDDF,PERSEC=N,HDRROW=Y,HDRPCOL=N,
//      DELIM=' ',DECSEP='.',
//      SAMPDIR=<db2ompe>.TK02SAMP,
//      FLDSSELS=<user_h1q>.FPECSVU(FPEPSDDF),
//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSDDF,
//      OUTDISP=OLD
// EXEC PROC=FPEPCSV,TYPE=SGBP,PERSEC=N,HDRROW=Y,HDRPCOL=N,
//      DELIM=' ',DECSEP='.',
//      SAMPDIR=<db2ompe>.TK02SAMP,
//      FLDSSELS=<user_h1q>.FPECSVU(FPEPSGBP),
//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSGBP,
//      OUTDISP=OLD
// EXEC PROC=FPEPCSV,TYPE=SGEN,PERSEC=N,HDRROW=Y,HDRPCOL=N,
//      DELIM=' ',DECSEP='.',
//      SAMPDIR=<db2ompe>.TK02SAMP,
//      FLDSSELS=<user_h1q>.FPECSVU(FPEPSGEN),
//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSGEN,
//      OUTDISP=OLD
// EXEC PROC=FPEPCSV,TYPE=SSET,PERSEC=N,HDRROW=Y,HDRPCOL=N,
//      DELIM=' ',DECSEP='.',
//      SAMPDIR=<db2ompe>.TK02SAMP,
//      FLDSSELS=<user_h1q>.FPECSVU(FPEPSSET),
//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSSET,
//      OUTDISP=OLD
// EXEC PROC=FPEPCSV,TYPE=SSTG,PERSEC=N,HDRROW=Y,HDRPCOL=N,
//      DELIM=' ',DECSEP='.',
//      SAMPDIR=<db2ompe>.TK02SAMP,
//      FLDSSELS=<user_h1q>.FPECSVU(FPEPSSTG),
//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSSTG,
//      OUTDISP=OLD
// EXEC PROC=FPEPCSV,TYPE= SXCL,PERSEC=N,HDRROW=Y,HDRPCOL=N,
//      DELIM=' ',DECSEP='.',
//      SAMPDIR=<db2ompe>.TK02SAMP,
//      FLDSSELS=<user_h1q>.FPECSVU(FPEPSXCL),
//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSXCL,
//      OUTDISP=OLD
// EXEC PROC=FPEPCSV,TYPE=SSIM,PERSEC=N,HDRROW=Y,HDRPCOL=N,
//      DELIM=' ',DECSEP='.',
//      SAMPDIR=<db2ompe>.TK02SAMP,
//      FLDSSELS=<user_h1q>.FPECSVU(FPEPSSIM),
//      INPUTDS=<user_h1q>.STLOAD,
//      OUTPUTDS=<user_h1q>.SCSVSSIM,
//      OUTDISP=OLD

```

You can now import the generated CSV data into your spreadsheet tools at the workstation and use graphical representations or pivot tables for performance analysis.

Chapter 26. Further reference information

The following information provides details about specific themes that are of interest on special occasions.

“Using GROUP to improve Save data set performance”

If you never require separate report entries for one or more OMEGAMON XE for DB2 PE identifiers, you can use GROUP to reduce the uniqueness of the key in the Save data. This can result in fewer reduced records in the Save data set, a smaller Save data set, and improved performance.

“Comparing original authorization IDs with primary authorization IDs” on page 286

This information explains the difference between original authorization ID and primary authorization ID.

“Fields affected by roll-up during distributed and parallel tasks” on page 287

The following table lists fields that are not reported by Accounting reports and traces if roll-up during distributed activity and parallel query tasks is active.

“Using ACCOUNTING REDUCE subcommand options to balance performance and data set space” on page 288

The subcommand options CALCULATE and OPTIMIZE estimate required ACWORK data set space and control the performance of Accounting report generation jobs that involve data aggregation by means of the REDUCE subcommand. The subcommand options are of interest if the reduction of high volume of Accounting data results in unacceptable performance, difficulties to determine adequate ACWORK space, or even abnormally terminated jobs (abend B37).

“Accessibility” on page 293

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use software products successfully.

“Troubleshooting empty reports” on page 296

This topic lists and explains several reasons why your report might not contain any report data. Use this information as a guide to identify possible reasons, including inappropriate command parameters.

Using GROUP to improve Save data set performance

If you never require separate report entries for one or more OMEGAMON XE for DB2 PE identifiers, you can use GROUP to reduce the uniqueness of the key in the Save data. This can result in fewer reduced records in the Save data set, a smaller Save data set, and improved performance.

The standard key for Save data sets contains a combination of the applicable OMEGAMON XE for DB2 PE identifiers from DB2 instrumentation records. For example, the key for an Accounting save record contains a number of OMEGAMON XE for DB2 PE identifiers. Because of the uniqueness of the standard key, there can be a very large number of reduced records in a Save data set.

If you never require separate report entries for one or more OMEGAMON XE for DB2 PE identifiers, for example, if you only produce Accounting reports by using ORDER(CONNECT-PRIMAUTH-PLANNAME), you can use groups in INCLUDE REDUCE to obtain the minimum number of records in the Save data set.

The following command stream does not use groups. The resulting Save data set contains reduced records for each unique combination of all of the OMEGAMON XE for DB2 PE identifiers in the key.

```

:
:
ACCOUNTING
  REDUCE
  SAVE
:
:

```

The following command stream uses groups and minimizes the number of records in the Save data set:

```

:
:
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 OMEGAMON XE for DB2 PE identifiers that are not grouped, for example, PRIMAUTH and PLANNAME.

During REDUCE processing, the group name is substituted for the original value of the key entry. For example, all records in the reduced data have a correlation name of ALLCNM.

Because the key now contains the group name (not the original value), you cannot use the original values of any of the grouped OMEGAMON XE for DB2 PE identifiers in REPORT or INCLUDE/EXCLUDE in the same command stream or when you restore the data.

Comparing original authorization IDs with primary authorization IDs

This information explains the difference between original authorization ID and primary authorization ID.

Original authorization ID

During connection to DB2 (either by IDENTIFY or SIGNON), an initial authorization value is passed to the connection exit. This value becomes the original authorization ID.

For IDENTIFY:

- If RACF® is active, this value is the verified user ID.
- If RACF is not active, this value is blank.

For IMS SIGNON:

- If RACF is active, this value is the terminal user ID.
- If RACF is not active, this value is either the LTERM name or the PSB name.

For CICS SIGNON:

- This value is determined by the user-defined CICS resource control table (RCT). The connection (authorization) exit can be either the IBM supplied default or user-written, depending upon whether secondary authorization IDs are used.

Primary authorization ID

The primary authorization ID is the value set by the exit. This value is determined according to the following criteria:

- Whether it is an IDENTIFY or a SIGNON
- Whether RACF is active or inactive
- Whether the exit is IBM supplied or user-written
- Whether secondary IDs are being used

Default values can be any of the following:

- The TSO logon ID
- The value of the USER field on the JOB statement
- A default value specified when you install DB2
- The original (unaltered) value

Note:

- The original authorization ID should be used when you attempt to establish accountability of DB2 activity, because the primary authorization ID can be an ID other than the user (group name, for example).
- If your subsystem uses authorization ID translation for distributed activity, the AUTHID reported for DBATs is the translated value.
- For detailed information about authorization IDs, see the *IBM Knowledge Center*.

Comparing secondary IDs with SQL ID

DB2 uses two other types of authorization IDs:

- Secondary authorization IDs
- SQL ID

If secondary IDs are used, a user-written authorization exit is also required. A secondary list can contain from 1 to 245 secondary IDs. This list is accessed when you establish the primary authorization ID or the SQL authorization ID.

The primary ID and the SQL ID are set during either IDENTIFY or SIGNON. However, only the SQL ID can be changed after connection by the SET CURRENT SQLID statement.

The SQL ID must be either the primary ID or one of the secondary IDs. It is used for implicit name qualifiers, implicit ownership assignment, and GRANT/REVOKE authorization checking.

For more detailed information about authorization IDs, see the *IBM Knowledge Center*.

Fields affected by roll-up during distributed and parallel tasks

The following table lists fields that are not reported by Accounting reports and traces if roll-up during distributed activity and parallel query tasks is active.

See “Distributed activity” on page 142 and “Query parallelism considerations” on page 143 for background information.

Table 28. Fields affected by roll-up

Field name	Field meaning
QPACANM	ACTIVITY NAME
QPACANM_VAR	ACTIVITY NAME
QPACASCH	SCHEMA NAME
QPACASCH_VAR	SCHEMA NAME
QPACBJST	CLASS 7: BEGINNING TCB CPU TIME
QPACCANM	STORED PROCEDURE EVENTS
QPACCAST	SCHED.PROCEDURE SUSP TIME
QPACCONT	CONSISTENCY TOKEN
QPACEJST	ENDING TCB CPU TIME
QPACSCB	BEGINNING STORE CLOCK TIME
QPACSCE	ENDING STORE CLOCK TIME
QPACUDST	SCHED.UDF SUSP TIME
QTXACLMT	LIMIT IN CPU 16 MICROSEC
QTXACHUS	HIGHEST CPU 16 MICROSEC USED
QTXAFLG1	RES LIMIT TYPE
QTXAPREC	RES LIMIT SCOPE
QTXARLID	RLF TABLE ID
QTXASLMT	LIMIT IN SERVICE UNITS
QWACNID	NETWORK ID VALUE

Using ACCOUNTING REDUCE subcommand options to balance performance and data set space

The subcommand options CALCULATE and OPTIMIZE estimate required ACWORK data set space and control the performance of Accounting report generation jobs that involve data aggregation by means of the REDUCE subcommand. The subcommand options are of interest if the reduction of high volume of Accounting data results in unacceptable performance, difficulties to determine adequate ACWORK space, or even abnormally terminated jobs (abend B37).

OMEGAMON XE for DB2 PE provides several commands for which the REDUCE subcommand and its options can be used to reduce the volume of data before the data is passed to subsequent other subcommands like REPORT and SAVE. When OMEGAMON XE for DB2 PE processes a REDUCE subcommand, it holds its temporary data in a REDUCE data set to relieve virtual storage. This temporary data set is either automatically allocated or, if its placement and size needs to be controlled, explicitly specified as a ccWORK DD statement in your JCL. (cc stands for the report set, respectively the command to which the temporary REDUCE data set is associated, for example, AC for the ACCOUNTING command).

The inherent shortcoming with REDUCE is the difficulty to determine in advance an appropriate size of the temporary data set. The required size depends mostly on the amount and complexity of input data to be reduced and on REDUCE subcommand options like FROM/TO, INCLUDE/EXCLUDE, INTERVAL, and BOUNDARY, which influence the reduction factor. In addition, the size of the

temporary data set has an inverse proportional effect on the processing time of a REDUCE step. These shortcomings might result in unacceptable performance of report generation jobs or even unexpected job terminations (B37 abends). These problems are more likely if large amounts of Accounting data are reduced.

Beginning with OMEGAMON XE for DB2 PE version 4.2, the REDUCE subcommand (if used with the ACCOUNTING command) provides the CALCULATE and OPTIMIZE subcommand options. The CALCULATE subcommand option provides information about ACWORK space actually used during a REDUCE step. If insufficient ACWORK space is specified, and the job consequently terminates, CALCULATE provides information about the size required to successfully complete the job in a second run. The OPTIMIZE subcommand option can be used to control the preference between performance and ACWORK space usage.

CALCULATE and OPTIMIZE are optional subcommand options. They can be used alone or in combination and specified in any sequence (logically, CALCULATE should be the last of several REDUCE options because they determine what needs to be calculated). Both options are purely performance- and processing-related and do not manipulate data in any way. For the latter purpose use REDUCE subcommand options like FROM/TO, INCLUDE/EXCLUDE, INTERVAL, and BOUNDARY.

The following sections describe the concepts and the appropriate uses of the CALCULATE and OPTIMIZE subcommand options. For the complete command syntax and for more information about the ACWORK DD statement, see the *Report Command Reference*.

Using ACCOUNTING REDUCE CALCULATE to tackle inappropriate ACWORK size allocations

This section describes how the use of the CALCULATE subcommand option helps to determine an adequate size for the ACWORK data set with regard to the amount and complexity of the data to be reduced.

By default, OMEGAMON XE for DB2 PE automatically allocates a temporary ACWORK data set to hold output data from the REDUCE subcommand. To control the placement and size of this data set, you can also specify a ddname of ACWORK. However, if the allocated size is too small for the temporary data, REDUCE processing terminates because of insufficient space (abend B37). Thereafter, you have to increase the size of the data set and run the job again, still without knowing an appropriate size.

You can use the CALCULATE subcommand option of REDUCE in the following manner in your OMEGAMON XE for DB2 PE command stream to force OMEGAMON XE for DB2 PE to calculate the appropriate size of ACWORK.

```
⋮  
ACCOUNTING  
  REDUCE  
    Other REDUCE subcommand options  
    CALCULATE  
  REPORT
```

SAVE
:

With CALCULATE in the command stream, OMEGAMON XE for DB2 PE counts the number and lengths of records written to ACWORK during ACCOUNTING REDUCE processing.

- In case ACWORK is sufficiently sized, REDUCE processing finishes normally and an information message in DPMLOG provides details about the calculations.

```
FPEA0800I THE ALLOCATED SPACE FOR ACWORK WAS SUFFICIENT.  
<V1> BYTES HAVE BEEN WRITTEN TO THE ACWORK DATA SET,  
CONSUMED BY <V2> RECORDS WITH AN AVERAGE LENGTH OF <V3>  
AND A MAXIMUM RECORD LENGTH OF <V4>.
```

Subsequent subcommands of ACCOUNTING (here, REPORT and SAVE) will be executed.

In case you specified the size of ACWORK, you can use the value shown as <V1> and compare it with your specification. You might want to adjust your specification to the same dimension, considering a bonus for variations of the input data.

- In case ACWORK is found to be too small, REDUCE processing stops writing to the data set. However, OMEGAMON XE for DB2 PE continues counting and calculating until all input data to REDUCE is processed. An error message in DPMLOG provides details about the minimum required size of ACWORK. Finally, the job terminates with abend B37.

```
FPEA0801S INSUFFICIENT SPACE WAS ALLOCATED FOR ACWORK.  
COMPLETELY REDUCED DATA WOULD REQUIRE AN ACWORK OF <V1> BYTES,  
CONSUMED BY <V2> RECORDS WITH AN AVERAGE LENGTH OF <V3>  
AND A MAXIMUM RECORD LENGTH OF <V4>.
```

Subsequent subcommands of ACCOUNTING (here, REPORT and SAVE) will not be executed.

In this case, you must specify the size of ACWORK to at least the value shown as <V1> (plus 20% is recommended). Then, you must rerun the job a second time.

In both messages the other values, shown as <V2>, <V3>, and <V4>, describe the amount and quality of processed records. The error message FPEA0801S is accompanied by other error messages indicating a failure to write to ACWORK. For more information, see *Messages*.

Using ACCOUNTING REDUCE OPTIMIZE to balance REDUCE performance

This section describes how the use of the OPTIMIZE subcommand option and its parameters helps to control CPU consumption in conjunction with required ACWORK space.

The REDUCE process takes place in virtual storage and requires several cycles to complete. By default, without the use of the OPTIMIZE subcommand option, the amount of virtual storage made available is dynamically adjusted for each cycle, based on the number of entries to be held in virtual storage for each cycle. This automatism can result in inappropriate allocation of virtual storage and in poor job performance.

With OPTIMIZE in use, constant thresholds for all cycles are used for the number of entries to be held in virtual storage, which results in a constant amount of

available virtual storage. The threshold values are internally derived and are adjusted based on the OPTIMIZE parameter provided by you.

In summary, a smaller amount of virtual storage holds fewer entries, which results in faster access, which decreases CPU consumption, but requires more ACWORK space. Vice versa, a larger amount of virtual storage holds more entries, which results in slower access, which increases CPU consumption, but requires less ACWORK space.

You can use the OPTIMIZE subcommand option as outlined below in your OMEGAMON XE for DB2 PE command stream to control whether the REDUCE process is performed with minimum CPU consumption (but maximum ACWORK space), minimum ACWORK space (but maximum CPU consumption), or any distinct value between these two boundary values.

```

:
ACCOUNTING
  REDUCE
    Other REDUCE subcommand options
    OPTIMIZE(parameter)
  REPORT
  SAVE
:

```

The OPTIMIZE subcommand option requires one of the following keywords or values. For more information about other ACCOUNTING subcommands and their options and interrelations, see the *Report Command Reference*.



The OPTIMIZE keywords and values have the following effects:

CPUTIME

The CPU consumption is minimized, at the expense of required ACWORK space. This setting is equivalent to OPTIMIZE(0).

ACWORK

The required ACWORK space is minimized, at the expense of CPU consumption. This setting is equivalent to OPTIMIZE(100).

n A numeric value from 0 to 100, specifying the balance between minimized CPU consumption (0) and minimized required ACWORK space (100).

The correlation between a specific value and a resource consumption (required ACWORK space) is not exactly linear. The reason is that any value from 0 to 100 that you specify is internally correlated to the maximum number of entries held in virtual storage. Nevertheless, this should be of limited interest for the sake of the intended usage.

The results from different OPTIMIZE settings, assuming identical input data to the REDUCE process, are nearly identical. Small differences should be considered normal. They result from different contents of ACWORK at different stages and from repeated multiplications of rounded values.

Example of fine-tuning by varying REDUCE OPTIMIZE values

This example shows one of several methods to fine-tune the data reduction process by monitoring the effects of different OPTIMIZE values for a given amount of data to be reduced. The goal is to find a range of OPTIMIZE values that results in acceptable CPU time and ACWORK space consumption.

To show noticeable variations in CPU time and ACWORK space consumption, it is assumed that the volume of data to be reduced is relatively high and the data is complex. Small trace samples might not show any effects. If your actual data does not result in noticeable variations, there is nothing to optimize.

In Table 29 on page 293 the resulting values for CPU time and ACWORK space consumption are derived from left to right.

1. The job is run with OPTIMIZE(0), which results in CPU time = 10 (acceptable, cannot be improved) and ACWORK = 50 (not acceptable).
Let's try the other extreme.
2. The job is run with OPTIMIZE(100), which results in CPU time = 35 (not acceptable) and ACWORK = 25 (acceptable, cannot be improved).
There might exist an OPTIMIZE(n) value between 0 and 100 that satisfies both criteria. Let's try the medium.
3. The job is run with OPTIMIZE(50), which results in CPU time = 25 (not acceptable) and ACWORK = 38 (acceptable).
OPTIMIZE(50) is still too high; we need to favor CPU time by using an OPTIMIZE(n) value between 0 and 50.
4. The job is run with OPTIMIZE(25), which results in CPU time = 15 (acceptable) and ACWORK = 45 (not acceptable).
OPTIMIZE(25) is too low; we need to favor ACWORK by using an OPTIMIZE(n) value between 25 and 50.
5. The job is run with OPTIMIZE(38), which results in CPU time = 19 (acceptable) and ACWORK = 42 (acceptable).
OPTIMIZE(38) satisfies both criteria. Let's try to favor ACWORK by using an OPTIMIZE(n) value between 38 and 50.
6. The job is run with OPTIMIZE(44), which results in CPU time = 23 (acceptable) and ACWORK = 41 (acceptable).
OPTIMIZE(44) still satisfies both criteria. Let's try again to favor ACWORK by using an OPTIMIZE(n) value between 44 and 50.
7. The job is run with OPTIMIZE(47), which results in CPU time = 24 (acceptable) and ACWORK = 39 (acceptable).
OPTIMIZE(47) still satisfies both criteria. Let's try again to favor ACWORK by using an OPTIMIZE(n) value between 47 and 50.
8. The job is run with OPTIMIZE(49), which results in CPU time = 24 (acceptable) and ACWORK = 39 (acceptable).
OPTIMIZE(49) still satisfies both criteria. In fact nothing has changed.

Based on the given workload and criteria, this example ascertains acceptable OPTIMIZE(n) values from 38 to 49.

Table 29. Protocol of REDUCE OPTIMIZE values and effects. OPTIMIZE value of 0 equals CPUTIME, OPTIMIZE value of 100 equals ACWORK.

OPTIMIZE value	0	100	50	25	38	44	47	49
Consumed CPU time [sec]	10	35	25	15	19	23	24	24
Acceptable?	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Consumed ACWORK space [MB]	50	25	38	45	42	41	39	39
Acceptable?	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Both acceptable?					Yes	Yes	Yes	Yes

You will find the actual values for consumed CPU time in the job output of JESMSGGLG (search for TOTAL CPU TIME=). The actual size of ACWORK can be determined through TSO option 3.2, or from messages FPEA0800I and FPEA0801S resulting from REDUCE CALCULATE.

The previous example also reveals the reverse effects of consumed CPU time and ACWORK space for ordered OPTIMIZE values.

Table 30. Effects of OPTIMIZE values on CPU time and ACWORK space

OPTIMIZE value	0	25	38	44	47	49	50	100
Consumed CPU time [sec]	10	15	19	23	24	24	25	35
Consumed ACWORK space [MB]	50	45	42	41	39	39	38	25

Accessibility

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use software products successfully.

This index describes the major accessibility features in OMEGAMON XE for DB2 PE, OMEGAMON XE for DB2 PM, and Buffer Pool Analyzer:

- You can operate all features using the keyboard instead of the mouse.
- You can change the system settings for high contrast for all user interface controls and client area contents.

The following sections explain how to use these accessibility features.

Operating all features by using the keyboard

You can use keys or key combinations to perform operations that can also be done through mouse actions. All menu items can be accessed from the keyboard. In those cases, the keyboard equivalent appears to the right of the menu item, or the shortcut letter is underlined. Some keyboard items also have shortcuts.

To navigate through a window or dialog by using the keyboard instead of the mouse, use the following keyboard shortcuts:

Table 31. Navigating through a window or dialog

Navigating through a window or dialog	Shortcut
Access the menu bar in a window	Alt+underlined letter or F10
Activate a menu item in the menu bar	Enter

Table 31. Navigating through a window or dialog (continued)

Navigating through a window or dialog	Shortcut
Access controls in a dialog	Alt+underlined letter
Navigate through the menu bar	Right arrow, left arrow, down arrow, up arrow
Move to the next set of controls	Tab or Ctrl+Tab
Move to the previous set of controls	Shift-Tab
Move within tables	Tab or right arrow, Shift-Tab or left arrow, down arrow, up arrow
Move within trees	Up arrow, down arrow
Expand a tree node	Right arrow
Collapse a tree node	Left arrow
Move within list boxes	Up arrow, down arrow
Move within combo boxes	Up arrow, down arrow
Move to a split bar in a window	F8
Resize a window by using the split bar	Arrow keys

To perform main tasks in windows by using the keyboard instead of the mouse, use the following keyboard shortcuts:

Table 32. Keyboard shortcuts in a window

Task	Shortcut
Display Help	F1
Close the active window	Ctrl+W or Alt+F4
Exit the application	Ctrl+Q
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Print	Ctrl+P
Refresh displayed data	F5
Enter or leave history mode	Ctrl+H
Move back in history mode	Alt+left arrow
Move forward in history mode	Alt+right arrow
Add a new subsystem in System Overview	Ctrl+N
Add a new configuration in Trace Configurations	Ctrl+N
Open a report in Buffer Pool Analyzer	Ctrl+O
Start a simulation process in Buffer Pool Analyzer	Ctrl+I
Cancel Thread in Thread Summary and Details	Del
Delete Trace Configuration in Trace Configurations	Del
Open Trace Configuration	Ctrl+Alt+N
Open Trace Activation	Ctrl+Alt+A
Open Exception Processing	Ctrl+Alt+E

Table 32. Keyboard shortcuts in a window (continued)

Task	Shortcut
Open System Overview	Ctrl+Alt+O
Open Buffer Pool Analyzer	Ctrl+Alt+B
Open DB2 command for the active subsystem	Ctrl+Alt+D
Open Statistics Details for the active subsystem	Ctrl+Alt+S
Open System Health for the active subsystem	Ctrl+Alt+H
Open All Threads in Lock Conflict for the active subsystem	Ctrl+Alt+L
Open Locking Conflicts for the active subsystem	Ctrl+Alt+C
Open System Parameters for the active subsystem	Ctrl+Alt+Y
Open Performance Warehouse for the active subsystem	Ctrl+Alt+P
Open Thread Summary for the active subsystem	Ctrl+Alt+T

To perform main tasks in dialogs by using the keyboard instead of the mouse, use the following keyboard shortcuts:

Table 33. Keyboard shortcuts in a dialog

Task	Shortcut
Confirm a dialog	Enter
Cancel a dialog	Esc
Activate a button that has the focus	Spacebar or Enter
Select and deselect check boxes and radio buttons	Spacebar
Navigate within check box and radio button groups	Arrow keys
Open combination box menu	Alt+down arrows
Close combination box menu	Esc
Move up and down in combination box menu	Up arrow, down arrow
Activate combination box menu item	Spacebar or Enter
Move within a list box	Up arrow, down arrow
Activate a list box entry	Enter
Move between the pages of a window that contains tabs if the tab has the focus	Right arrow, left arrow

Changing the system settings for high contrast and font size

You can change the system settings for high contrast and font size.

To use specific settings that are designed for easy reading, take these steps:

1. In the OMEGAMON XE for DB2 PE - System Overview window, click **Monitor** → **Configuration**.
2. In the Configuration window, click **Accessibility**.

3. To specify the appearance scheme for high contrast and font size, select one of the following options:
 - To use the default settings, select **[None]**.
 - To specify a white background containing black text in normal font size, select **Black on white**.
 - To specify a white background containing black text in large font size, select **Black on white (large font)**.
 - To specify a black background containing white text in normal font size, select **White on black**.
 - To specify a black background containing white text in large font size, select **White on black (large font)**.

You have to restart the application before the new settings come into effect.

Troubleshooting empty reports

This topic lists and explains several reasons why your report might not contain any report data. Use this information as a guide to identify possible reasons, including inappropriate command parameters.

At its core, OMEGAMON XE for DB2 PE identifies data to be collected and reported by IFCIDs. “DB2 instrumentation data” on page 24 provides an overview of how IFCIDs map to report sets, DB2 trace types and DB2 trace classes. The basic rule is that data to be reported needs to be collected and made available as input data to the report generation process. If required IFCIDs are missing, the generated report ends with NO DATA TO REPORT.

The problem of missing IFCIDs might also be caused by different versions of DB2 or OMEGAMON XE for DB2 PE. Both products might support different sets of IFCIDs, which can lead to situations where an IFCID is either not provided by a DB2 version or cannot be processed by an OMEGAMON XE for DB2 PE version. The same problem might arise if data from other sources, like SMF or GTF, is used for reports.

OMEGAMON XE for DB2 PE provides flexible means to specify which data to collect and which data to report. However, it provides no means to protect against inappropriate use of commands, subcommands and subcommand options, which also can result in empty reports.

The following sections list possible reasons why your report might end with a final message of NO DATA TO REPORT.

Messages indicating unsupported IFCIDs or IFCID formats

If the job summary log shows message FPEC4015I NUMBER OF RECORDS FROM UNSUPPORTED RELEASES OF DB2 WAS ..., OMEGAMON XE for DB2 PE has detected IFCIDs in its input data that it is not supposed to handle. Either the input data is from an outdated DB2 version and the IFCID is no longer supported by OMEGAMON XE for DB2 PE, or the input data contains IFCIDs from a newer DB2 version and OMEGAMON XE for DB2 PE is not yet aware of them.

If the job summary log shows message FPEC4020I NUMBER OF RECORDS FROM UNSUPPORTED PRODUCT RELEASES WAS ..., the input data (such as input data in DPMOUT format) was created by earlier versions of OMEGAMON XE for DB2 PE or OMEGAMON XE for DB2 PM and is no longer supported.

Identifying missing IFCIDs

Report sets such as Accounting, Locking, or Statistics require specific IFCIDs from DB2 instrumentation data, as outlined in “DB2 instrumentation data” on page 24. If the input data sets (specified with the INPUTDD statement in your job stream) do not contain these IFCIDs, the generated report ends with NO DATA TO REPORT.

For example, an Accounting report requires IFCIDs 3 and 239 in the input data. If the job summary log shows nothing more than in the following example (no IFCIDs 3 and 239), your Accounting report remains empty.

IFCID	INPUT COUNT	INPUT PCT OF TOTAL	PROCESSED COUNT	PROCESSED PCT OF TOTAL
1	89	33.58%	0	0.00%
2	88	33.20%	0	0.00%

Note: The job summary log gives a clear indication about which IFCIDs are contained in your input data. If an IFCID is not listed in the IFCID column or the INPUT COUNT column shows a count of 0, your input data does not contain this IFCID.

Reasons for missing IFCIDs

IFCIDs might be missing in your reports because they are either not collected (missing in the input data for the report generation) or being accidentally suppressed or filtered by inappropriate use of commands, subcommands, or subcommand options.

To identify why specific IFCIDs are not collected, review the methods of how the data was collected. Chapter 3, “Collecting data for reports,” on page 9 describes several methods, some of them let you explicitly specify what to collect. For example, the DB2 startup parameters might determine that certain trace classes (with their implied IFCIDs) are not started.

If your input data to specific reports contains the required IFCIDs, but your reports still do not show any data, it is likely that restrictive command or subcommand options are in use in your job stream.

- The GLOBAL command might apply global filters to all data serving as input to subsequent OMEGAMON XE for DB2 PE commands like ACCOUNTING or STATISTICS.
- OMEGAMON XE for DB2 PE commands like ACCOUNTING or STATISTICS might use subcommand options that further restrict the data. The most important ones are:
 - FROM/TO might limit the time frame too much.
 - INCLUDE/EXCLUDE might limit a DB2 trace class or an IFCID.

For more information about commands and subcommand options, see the *Report Command Reference*.

For example, a Statistics report requires IFCIDs 1 and 2. The following job summary log shows that 48 records of each IFCID are included in the input data. However, none of these are being processed (PROCESSED COUNT equals 0) during the report generation and consequently the Statistics report is empty.

IFCID	INPUT COUNT	INPUT PCT OF TOTAL	PROCESSED COUNT
1	48	25.00%	0
2	48	25.00%	0

Insufficient amount of Statistics record pairs (IFCIDs 1 and 2)

Output in a Statistics report requires at least two DB2 Statistics record pairs (IFCIDs 1 and 2) in the input data. These IFCIDs are collected at a configurable interval (DB2 system parameter STATIME). For more information, see Chapter 18, “General Statistics information,” on page 189.

An empty Statistics report usually indicates that either the input data does not cover at least two intervals or that filters restrict the amount of data being processed during report generation.

The following example of a job summary log shows that a sufficient amount of IFCID 1 and 2 records are in the input data (INPUT COUNT column). However, filters have limited the number of processed records below the required minimum of two record pairs (PROCESSED COUNT column). Consequently, the Statistics report is empty.

IFCID	INPUT COUNT	INPUT PCT OF TOTAL	PROCESSED COUNT	PROCESSED PCT OF TOTAL
1	48	25.00%	1	1.02%
2	48	25.00%	1	1.02%

The following job stream example shows how improper use of subcommand options cause an empty Statistics report. Here, the REDUCE step causes a summarization of input data to an interval of 60 minutes, with a boundary to the start of the hour. However, in the REPORT step the FROM and TO subcommand options limit the data to 30 minutes, which is less than a single interval.

```

:
:
STATISTICS
  REDUCE
    INTERVAL(60)
    BOUNDARY(60)
  REPORT
    FROM (12/22/13,12:15:00.00)
    TO   (12/22/13,12:45:00.00)
:
:

```

Messages indicating incomplete Accounting data (IFCIDs 3 and 239)

If the job summary log shows one or more of the following messages, incomplete Accounting data was found in the input data and your Accounting report might be incomplete. In the worst case, the report might end with NO DATA TO REPORT.

- FPEA4531I GENERAL ACCOUNTING DATA FOR LUWID INSTANCE <V1> IS MISSING. IFCID 3 IS MISSING.
- FPEA4532I ACCOUNTING DATA FOR A NUMBER OF PACKAGES FOR LUWID <V1> IS MISSING. ONE OR MORE IFCID 239 IS MISSING.
- FPEA4534I COORDINATING PARALLEL TASK FOR LUW INSTANCE <V1> IS NOT REPORTED AS NOT ALL INFORMATION FOR ASSISTING PARALLEL TASKS FROM MEMBER <V1> IS AVAILABLE.

Accounting reports require IFCID 3 for reporting a thread. However, package or program data belonging to a thread is contained in multiple 239 IFCIDs. A 1: n relation is established within the data. For correct reporting a complete set of these IFCIDs is required in the input data.

In case of DB2 Query Parallelism, additional sets of IFCIDs 3 and 239 are required. They were created from parallel processors or from DB2 systems that assisted an initiating agent. Dependent on the degree of parallelism, this can result in a 1: p relation between agent and parallel tasks. In other words, correct Accounting processing needs from the agent one IFCID 3 and n IFCIDs 239, and for each of the p parallel tasks also one IFCID 3 and a certain number of IFCIDs 239.

Accounting processing needs a complete set of these IFCIDs for a correct reporting. If either IFCIDs 239 belonging to an IFCID 3 are missing, or if IFCIDs 239 are available but their IFCID 3 is missing, the entire thread is excluded from further processing and reporting.

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 grant 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 character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland GmbH
Dept. M358
IBM-Allee 1
71139 Ehningen
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 document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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 measurements 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 illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com/legal/us/en/copytrade.shtml) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/us/en/copytrade.shtml>.

Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Bibliography

You can order many IBM publications such as product manuals or IBM Redbooks online at the *IBM Publications Center* website.

You can also order by telephone by calling one of the following numbers:

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

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

IBM Tivoli OMEGAMON XE for DB2 Performance Expert publications

The product library for Version 5 Release 3 covers the following information units:

OMEGAMON XE for DB2 PE and OMEGAMON XE for DB2 PM

- *Configuration and Customization*, GH12-7054
- *Parameter Reference*, SH12-7055
- *Monitoring Performance from the OMEGAMON Classic Interface*, SH12-7050
- *Monitoring Performance from the IBM Tivoli OMEGAMON Enhanced 3270 User Interface*, SH12-7056
- *Monitoring Performance from Performance Expert Client*, SH12-7051
- *Monitoring Performance from ISPF*, SH12-7052
- *Report Command Reference*, SH12-7048
- *Report Reference*, SH12-7047
- *Reporting User's Guide*, SH12-7053
- *Messages*, GH12-7049
- *Program Directory for Performance Monitor*, GI19-5016
- *Program Directory for Performance Expert*, GI19-5014
- *Quick Start Guide for the SQL Dashboard and the end-to-end SQL monitoring functions*, GH12-7046

Buffer Pool Analyzer

- *Buffer Pool Analyzer Configuration Guide*, SH12-7058
- *Buffer Pool Analyzer User's Guide*, SH12-7057
- *Program Directory for IBM DB2 Buffer Pool Analyzer for z/OS*, GI19-5017

InfoSphere Optim Performance Manager for Linux, UNIX, and Windows

- *InfoSphere Optim Performance Manager Installation Guide*, GC19-2934

The documentation is provided in PDF and htm format in the:

- *Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS Knowledge Center*
- *Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS Knowledge Center*

IBM Tivoli Monitoring publications

For the most current list of publications, see *Tivoli Monitoring in the IBM Knowledge Center*.

IBM DB2 publications

For the most current list of publications, see *IBM DB2 Tools Product Page*.

Other IBM publications

For IBM publications that are not directly related to OMEGAMON XE for DB2 PE and PM, see *IBM Publications Center* or *IBM Knowledge Center*.

Index

A

- access path 85
- accessibility 293
- accessibility features xv
- Accounting 57
 - class 1 times 148
 - class 2 times 148
 - DB2 trace 24
 - description 5
 - distributed activity 142
 - distributed data 215
 - distributed times 146
 - effective reports 151, 158, 160, 171, 173, 185
 - exception processing 158
 - exception report 71
 - general 137
 - grope-scope report 223
 - input for reports and traces 148
 - MAINPACK identifier 252, 257
 - missing data 150
 - nondistributed times 145
 - ORDER processing 160
 - processing considerations 151
 - query parallelism 143
 - record generation 149
 - tailoring report and trace layouts 236, 245
 - TOP processing 171
 - trace classes 145
 - when records are written 149
 - where information comes from 145
- Accounting classes 149
- Accounting report
 - rolled up values 143, 144
- Accounting report (short)
 - generating
 - with ISPF/PDF 36
 - tailoring 236
- Accounting trace
 - rolled up values 143, 144
- ACCUMACC
 - subsystem parameter 143
- accumulated value 190
- Accumulated value (delta) 191
- Accumulated value (interval) 192
- aggregation 138
- allied thread 139
- allied-distributed thread 140
 - definition 217
- audit
 - description 6
 - trace 20
- Audit trace
 - starting automatically 20
- AUTHID/PRIMAUTH identifier
 - comparing original with primary 286, 287
 - DB2 START TRACE command 22
- authorization
 - explain 87

- auxiliary command
 - description 34

B

- batch processing
 - description 3
- batch report
 - creating 31
- bibliography 305
- block of data
 - adding 239
 - replacing fields 241
- Block Selection panel 238
- buffer pool efficiency 52

C

- CALCULATE subcommand option of REDUCE 288
- CASE command 34
- class 3 suspensions
 - server 148
- CLASS option 22
- class, trace 24
 - description 24
- classes, Accounting 149
- classes, Accounting trace 145
- classes, trace 149
- Collect Report Data (CRD) panels
 - access to 12
- Collect Report Data Batch
 - program 27
- Collect Report Data Batch program
 - syntax 27
- Collect Report Data facility
 - collecting data for batch processing 10
 - DB2 START TRACE command 20, 24
 - INSTALL parameter values 20
 - using IFCIDs 24
- Collect Report Data panel 12
- collect task
 - configuring 13
 - restricting data types 14
 - restricting identifiers 16
 - restricting IFCIDs 15
 - starting and stopping traces 17
- collecting data for batch processing 11
- collecting data with Near-Term History
 - Data Collector 30
- command
 - ACCOUNTING
 - optimizing REDUCE performance 288
 - CASE 34
 - description 34
 - FIELD 34
 - for the individual report sets 34
 - GLOBAL 34

- command (*continued*)

- GROUP 34
 - defining OMEGAMON XE for DB2 PE identifier values 231
 - improving Save data set performance 285
- LIST 34, 232
- START TRACE 20, 24
 - understanding syntax diagrams xii
- comments, sending xvi
- configuring a collect task 13
- conventions used ix
- correlation identifier
 - default translation 251
- correlation identifier, default
 - translation 251
- correlation name
 - default translation 251
- correlation number, default
 - translation 251
- correlation translation 251
 - changing defaults 252
 - defaults 251
- CORRNAME identifier
 - default translation 251
- CORRNMBR identifier
 - default translation 251
- CP parallelism 143
- CPU clock setting, example 249, 250
- CRD (Collect Report Data) panels
 - access to 12
- CSV data 275
- customizing
 - Accounting and Statistics reports and traces 236, 245

D

- data definition name (ddname) 31, 34
- data set
 - ddnames 31, 34
 - DPMOUT
 - description 32
 - streamlining processing 233
 - for the individual report sets 33
 - general 32
 - producing 4
- data set maintenance menu
 - defining exception thresholds 66
 - defining the MAINPACK identifier 253
 - specifying time zones 246
- data sharing 223
 - collecting input 223
 - monitoring entire groups 221, 222
 - monitoring individual members 219, 220
- data sharing group 198
- Data to Collect window 14
- data, DDF 140
- database access thread 140

- DB2 instrumentation facility
 - description 8
 - START TRACE command 20, 24
- DB2 Statistics record pair 190
- DBAT 140
- DBAT-distributed thread 140, 218
- DDF (Distributed Data Facility) 215
 - between DB2 and non-DB2 systems 217
 - between DB2 systems 216
 - description 215
 - examples 216
 - selecting threads 216
- DDF data 140
- ddname
 - ACWORK
 - determine size for REDUCE processing 288
- ddname (data definition name) 31, 34
- defining
 - Exception Log data set in JCL 66, 158
 - Exception Log File data set in JCL 66, 159, 209
- definition
 - Exception File data set 66
 - exception log 66
 - exception report 65
- delta record 191
- DEST option 21
- DGOBFLD3 panel 241
- DGOBFLD4 panel 240
- DGOBMENU panel 237
- DGOBPBLK panel 238
- DGOBPLAY panel 238
- DGOBPSAV panel 244
- DGOETV51
 - different versions of 209
 - sample Exception Threshold data set member 209
 - warning message 209
- DGOFEPO0 panel 211
- DGOMAP00 panel 12
- DGOMAP10 window 19
- DGOMAP30 window 13
- DGOMAP31t window 14
- DGOMAP32 window 15
- DGOMAP33 window 16
- DGOMPA40 window 17
- DGOOJOBM panel 212
- DGOPMENU panel
 - defining exception thresholds 66
 - defining the MAINPACK identifier 253
 - specifying time zones 246
- DGOPPLDS panel 246
- DGOPPMDS panel 253
- DGOPXDS1 panel 67
- DGOPXDS2 panel 68
- DGOPXDSN panel 68, 72
- disability 293
- distributed Accounting times 146
- distributed activity 142
 - effects on roll-up 143
- distributed transaction 141
- DPMLOG data set 32

- DPMLOG execution log
 - description 4
- DPMOUT data set
 - description 32
 - streamlining processing 233
- DPMPARMS data set 32

E

- empty reports 9, 296
- ERRDMPDD data set 32
- exception
 - specifying 66
- exception field
 - changing the description 259
- Exception File data set
 - definition 66
- exception log 158
 - definition 66
 - description 4
- Exception Log data set
 - defining in JCL 66, 158
- Exception Log File data set
 - attributes 209
 - defining in JCL 66, 159, 209
- exception processing 207
 - Accounting 158
 - description 65
 - output types 65, 207
 - profiling 210
 - statistics 202
 - which thresholds to specify 73
- exception profiling
 - data sets involved 210
 - when to use 74
- Exception Profiling panel 211
- exception report 158
 - Accounting 71
 - definition 65
- exception threshold 158
 - streamlining processing 232
- Exception Threshold Category Selection panel 67
- Exception Threshold data set
 - attributes 209
 - block size 209
 - different versions of 209
 - modifying 209
 - purpose 209
 - record format 209
 - record length 209
 - sample 73
 - sample member 72
 - sample RKO2DATA(DGOETV51) 209
 - warning message 209
- Exception Threshold Field Details panel 68, 72
- Exception Threshold Field Selection panel 68
- EXCLUDE subcommand option 228
 - filtering input 228
- EXCPTDD data set 33
- EXFILDD1 data set 33
- explain 85
 - authorization 87
 - data set 34
 - description 7

- explain (*continued*)
 - distributed data 215
 - streamlining processing 233
- EXTRCDD1 data set 33

F

- field selection panel 241
- FILE subcommand 35
 - description 35
- filtering input 226, 229
- FPEA0800I message 290
- FPEA0801S message 290
- FROM subcommand option 226
 - filtering input 226

G

- Generation Data Group (GDG) 27
- GLOBAL command
 - description 34
 - suppressing PRESORTED 230
 - TIMEZONE option 245, 250
- GROUP command
 - defining OMEGAMON XE for DB2 PE identifier values 231
 - description 34
 - improving Save data set performance 285
- group-scope report 198, 219
 - Accounting 223
 - collecting input 223
 - description 219
 - example 221
 - Statistics 222
- group-scope trace 219
 - description 219
- group, data sharing 198

H

- headers
 - statistics 204
- high water mark value 191
- historical data 233
 - streamlining processing 233

I

- I/O activity
 - description 6
 - input 104
- identification, thread 140
- IFCID (instrumentation facility component identifier)
 - description 24
 - frequency distribution log description 4
- IFCID 350
 - for SQL Activity trace 79, 82
- IFCID 63
 - for SQL Activity trace 82
 - SQL statement text 82
- IFCID option 22
- IFCID Selection window 15

- INCLUDE subcommand option 228
 - filtering IFCIDs 85
 - filtering input 228
- input
 - I/O activity 104
 - SQL activity 77
- INPUTDD data set 32
- INSTALL parameter values 20
- instrumentation facility
 - description 8
 - START TRACE command 20, 24
- Interactive Report Facility (IRF)
 - Accounting Report panel 44
 - Accounting Report Selections panel 44
 - Background DDname Selections panel 45
 - composing a batch report command stream 41
 - example of generated job stream 46
 - example of producing an Accounting report 42
 - foreground and background processing 41
 - Interactive Report Selections panel 39, 42
 - Job Processing Selections panel 46
 - overview 39
 - Recall Selections panel 42
 - saving and recalling selections 42
 - selecting report sets 40
 - Session Options window 43
 - specifying default options 41
 - using BROWSE command 40
 - using COMPOSE command 42
 - using EXCLUDE command 40
 - using GROUP command 40
 - using INCLUDE command 40
 - using LIST command 40
 - using OPTIONS command 40
 - using RECALL command 40
 - using RESET command 40
 - using SAVE command 40
- interval (statistics) 189
- INTERVAL option 232
 - streamlining processing 232
- interval record 191
- introduction to OMEGAMON XE for DB2 PE 1
- IRF
 - see Interactive Report Facility (IRF) 39
- IRF Save Selections data set
 - attributes 42
- ISPF Online Monitor
 - access CRD panels 12
 - access to Collect Report Data panels 12
 - using 11
- ISPF/PDF editor, using 36

J

- JESMSGLG 293
- Job Processing Selections panel 212
- job summary log
 - description 4

- JOBSUMDD data set 32
- JSSRSD (job summary VSAM data set) 32

L

- layout
 - adding a block 239
 - displaying 243
 - replacing fields 241
 - saving 244
 - selecting 237
 - tailoring Accounting and Statistics reports and traces 236, 245
- LAYOUT subcommand option 230
- limiting input 226, 229
- LIST command 34, 232
 - description 34, 232
- LOCATION parameter 23
- location, requester 143
- locking
 - description 6
 - group-scope report 221
 - member-scope report 219
- log, exception 158
- logs
 - description 4

M

- MAINPACK Definition Member Editor panel 253
- MAINPACK identifier
 - defining 252, 257
 - examples 255, 256
- manage DB2 traces 8
- member scope 138
- member-scope report 196, 219
 - description 219
 - example 219
- member-scope trace 219
 - description 219
- menu
 - data set maintenance
 - defining exception thresholds 66
 - defining the MAINPACK identifier 253
 - specifying time zones 246
- message
 - DPMLOG execution log 4
 - FPEA4531I 298
 - FPEA4532I 298
 - FPEA4534I 298
 - FPEC4015I 296
 - FPEC4020I 296
 - NO DATA TO REPORT 296
- monitoring data sharing groups
 - entire group 221, 222
 - individual members 219, 220
- multi-site report 142

N

- Near-Term History Data Collector
 - using sequential data sets 30

- non-DDF data
 - definition 139
- nondistributed Accounting times 145
- nondistributed transaction 141

O

- OMEGAMON XE for DB2 PE
 - introduction 1
 - output types 4
- OMEGAMON XE for DB2 PE identifier AUTHID/PRIMAUTH
 - comparing original with
 - primary 286, 287
 - DB2 START TRACE command 22
- defining sets 231
- filtering input using INCLUDE and EXCLUDE 228
- MAINPACK 252, 257
- ORIGAUTH 286, 287
- OPTIMIZE subcommand option of REDUCE 288
- ORDER processing
 - Accounting 160
- Order subcommand option
 - in distributed environment 162
 - ORDER (ACTNAME) 166
 - ORDER (CONNECT) 161
 - ORDER (CORRNAME) 162
 - ORDER (INTERVAL) 170
 - ORDER (PACKAGE) 166
 - ORDER (PLANNAME-MAINPACK) 164
 - ORDER (PLANNAME) 164
 - ORDER (PRIMAUTH-PLANNAME) 161
- ORIGAUTH identifier 286, 287
 - comparing with primary authorization ID 286, 287
- original authorization ID 286
- output data set for DB2 trace data
 - attributes 18

P

- pairs, record 193
- panel
 - Accounting Report 44
 - Accounting Report Selections 44
 - Background DDname Selections 45
 - Collect Report Data 12
 - Exception Profiling 211
 - Exception Threshold Category Selection 67
 - Exception Threshold Field Details 68, 72
 - Exception Threshold Field Selection 68
 - Interactive Report Selections 39, 42
 - Job Processing Selections 46, 212
 - MAINPACK Definition Member Editor 253
 - Recall Selections 42
 - Time Zone Data Editor 246
 - User-Tailored Reporting Layout Generation 237

- panel (*continued*)
 - UTR Block Selection 238
 - UTR Field Selection 240, 241
 - UTR Layout Selection 238
 - UTR Save Layout Details 244
- parallelism
 - effects on roll-up 144
- parallelism, query CP 143
- parallelism, Sysplex query 143
- parameter
 - INSTALL 20
 - LOCATION 23
- performance data
 - generating, collecting, processing 8
 - using the workstation client 8
- performance data collection 9, 24
 - DB2 START TRACE command 20, 24
 - INSTALL parameter values 20
 - using IFCIDs 24
 - using Performance Expert Client 10
- performance database 261
 - Accounting tables 263
 - audit tables 265
 - exception table 267
 - locking tables 267
 - Record Trace tables 268
 - Statistics table 270
- performance warehouse 261
- PLAN option 22
- PRESORTED option
 - suppressing 230
- PRESORTED option, suppressing 230
- PRIMAUTH/AUTHID identifier
 - comparing original with primary 286, 287
 - DB2 START TRACE command 22
- processing exceptions 207
- program
 - Collect Report Data Batch 27
- PTASKROL
 - subsystem parameter 144

Q

- query CP parallelism 143
- query parallelism 143
- query parallelism, Sysplex 143

R

- record pair, DB2 Statistics 190
- record pairs 193
- record trace
 - description 7
- record, delta 191
- record, interval 191
- redbooks xiv
- REDUCE subcommand 35
 - calculate ACWORK size 288
 - control performance 288
 - description 35
- report
 - accounting
 - controlling the level of detail 230
 - Accounting report example, using IRF 42

- report (*continued*)
 - composing JCL with IRF 41
 - controlling the level of detail 230
 - creating 31
 - description 4
 - empty 9
 - grope-scope 222, 223
 - input 148
 - selecting report set 39
 - statistics
 - controlling the level of detail 230
- report set
 - commands 34
 - description 4
 - generating
 - using ISPF/PDF 36
 - level of detail 7
 - reporting process overview 8
- REPORT subcommand 35
 - description 35
- report, group-scope 198
- report, member-scope 196
- report, multi-site 142
- report, single-site 142
- report, user-tailored 236
- reports and traces
 - input for Statistics 193
- requester location 143
- resource access
 - tracking 121
- RESTORE subcommand 36
 - description 36
- RKD2PAR(COPT<ssid>)
 - collection options 30
 - WRITEOPTION keyword 30
- RKO2DATA data set
 - DOGETV51 member 72, 73
- RKO2DATA(DGOETV51)
 - different versions of 209
 - sample Exception Threshold data set 209
 - warning message 209
- roll-up
 - controlling 143, 144

S

- sample 27
- Save data set
 - improving performance 285
- Save Layout Details panel 244
- SAVE subcommand 36, 139, 194
 - with CONVERT option 36, 139, 194
 - without CONVERT option 36, 139, 158, 194
- scope, member 138
- sending comments xvi
- service xv
- short report
 - accounting
 - generating with ISPF/PDF 36
 - tailoring 236, 245
 - single-site report 142
- SMF (system management facility) 21
- snapshot value 191
- snapshot value (delta) 191
- snapshot value (interval) 192

- Spreadsheet Input-Data Generator 275
- SQL activity 75
 - description 6
 - distributed data 215
 - input 77
- SQL Activity trace
 - description 6
 - IFCID 63 82
 - SQL statement text 82
- SQL statement 85
- SQL statement text
 - IFCID 350 82
 - IFCID 63 82
 - SQL Activity trace 82
- START TRACE
 - collecting IFCID 350 83
- START TRACE command 20, 24
 - syntax 20
- starting and stopping traces of a collect task 17
- statistics
 - description 5
 - distributed data 215
 - exception processing 202
 - grope-scope report 222
 - headers used 204
 - how values are reported 205
 - reporting values 205
 - tailoring report and trace layouts 236, 245
 - values reporting 205
- Statistics 49
 - functions 193
 - input for reports and traces 193
 - level of detail 195
 - sample layouts 195
- Statistics record pair 190
- STEPLIB data set 32
- subcommand 35
 - description 35
 - FILE 35
 - REDUCE 35
 - using CALCULATE option 288
 - using OPTIMIZE option 288
 - REPORT 35
 - RESTORE 36
 - SAVE 36, 139, 194
 - TRACE 35
 - with CONVERT option 36, 139, 194
 - without CONVERT option 36, 139, 194
- subsystem parameter
 - ACCUMACC 143
 - PTASKROL 144
- support home website xv
- suspensions, class 3 148
- syntax diagrams xii
- SYSIN data set 32
- SYSOUT data set 32
- Sysplex query parallelism 143
- system management facility (SMF) 21
- system overview 49
- system parameters reports
 - data set 34
 - description 7

T

- tailoring
 - Accounting and Statistics reports and traces 236, 245
- TDATA option 22
- terminology online xv
- terminology used xii
- thread 57
 - selecting for DDF 216
- thread identification 140
- thread types 139
- thread, allied 139
- thread, allied-distributed 140
- thread, database access 140
- thread, DBAT-distributed 140
- threshold
 - types 158
- threshold, exception 158
- time zone
 - editor 246
 - example 248, 250
 - specifying 245, 250
- Time Zone Data Editor panel 246
- times, distributed Accounting 146
- times, nondistributed Accounting 145
- TIMEZONE option 245, 250
 - GLOBAL command 245, 250
- TKO2SAMP(FPEZCRDJ) 27
- TO subcommand option 226
 - filtering input 226
- TOP (ONLY) subcommand option 160
- TOP list
 - producing 171
- TOP processing
 - Accounting 171
- trace
 - accounting
 - controlling the level of detail 230
 - controlling the level of detail 230
 - description 4
 - input 148
 - starting automatically 20
 - statistics
 - controlling the level of detail 230
- trace class
 - description 24
- trace classes 149
 - 30, 31, 32 83
- trace classes, Accounting 145
- Trace Configuration window 13
- TRACE option 21
- Trace Qualification window 16
- Trace Status Summary window 19
- TRACE subcommand 35
 - description 35
- traces and reports
 - input for Statistics 193
- traces, starting and stopping 17
- transaction, distributed 141
- transaction, nondistributed 141
- Trigger by Time window 17
- troubleshooting 296

U

- updates xv
- user-tailored report 236
- User-Tailored Reporting Layout
 - Generation panel 237
- using the ISPF Online Monitor 11
- using with Collect Report Data Batch program 27
- utility
 - Spreadsheet Input-Data Generator 275
- utility activity
 - description 6
- UTR (User-Tailored Reporting) 236, 245
 - Accounting report (short) 236
 - block of data
 - adding 239
 - replacing fields 241
 - description 236
 - layout
 - displaying 243
 - saving 244
 - selecting 237
- UTR Block Selection panel 238
- UTR field selection panel 241
- UTR Field Selection panel 240
- UTR Layout Selection panel 238
- UTR Save Layout Details panel 244

V

- value
 - GROUP command 231
 - INSTALL parameter 20
- value, accumulated 190
- value, Accumulated (delta) 191
- value, Accumulated (interval) 192
- value, high water mark 191
- value, snapshot 191
- value, snapshot (delta) 191
- value, snapshot (interval) 192

W

- where to find information xiv
- window
 - Data to Collect 14
 - DGOMAP10 19
 - DGOMAP30 13
 - DGOMAP31 14
 - DGOMAP32 15
 - DGOMAP33 16
 - DGOMPA40 17
 - IFCID Selection 15
 - Session Options 43
 - Trace Configuration 13
 - Trace Qualification 16
 - Trace Status Summary 19
 - Trigger by Time 17
- workload 75
- Workstation Online Monitor
 - manage DB2 trace 10
- WRITEOPTION keyword 30

Readers' Comments — We'd Like to Hear from You

IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS
IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS
Reporting User's Guide
Version 5.3.0

Publication No. SH12-7053-01

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Send your comments to the address on the reverse side of this form.

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

Email address



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Research & Development GmbH
User Experience & 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



Product Number: 5655-W37
5655-W38

SH12-7053-01

