IBM DB2 SQL Performance Analyzer for z/OS
Version 4  Release 2

User's Guide

IBM
## Contents

**About this information** ........................................ vii
  How to read syntax diagrams .................................. vii
  How to send your comments .................................. vii

**Part 1. Product overview** .................................. 1

  **Chapter 1. DB2 DB2 SQL Performance Analyzer overview** .......... 3
  What’s new in DB2 SQL Performance Analyzer .......................... 3
  What does DB2 SQL Performance Analyzer do? ........................ 4
  DB2 SQL Performance Analyzer features and benefits ................... 4
    Generic plan tables ........................................ 5
    Catalog access ............................................ 5
    DB2 SQL Performance Analyzer and QMF ........................... 5
    Remote operation through DRDA ............................... 6
    Accelerated table and SQL report processing ....................... 6
    DB2 SQL Performance Analyzer stored procedure capability .......... 6
    SQL Advisor ............................................. 7
    Exclude WHERE CURRENT OF statements from Query Limit reports ...... 7
    Actual query cost ........................................ 7
  DB2 SQL Performance Analyzer components and facilities ................ 11
    Parser and catalog interfaces ................................ 11
    Resource estimator ....................................... 11
    QMF Intercept .......................................... 12
    CAF and QMF features that enhance SQL PA ....................... 12
    Easy Explain .......................................... 13
    Tools Customizer overview ................................ 14
  DB2 SQL Performance Analyzer architecture and process flow .......... 14
  DB2 SQL Performance Analyzer operating environments .................. 15
  Service updates and support information ............................ 17
  Product documentation and updates ................................ 17
  Accessibility features ....................................... 18

**Part 2. Customization** ..................................... 21

  **Chapter 2. Customizing DB2 SQL Performance Analyzer** .......... 23
  Preparing to customize DB2 SQL Performance Analyzer ................. 23
    Set up your environment prior to customization ............... 25
    Plan table configurations for DB2 EXPLAIN output .......... 26
    Requirements for using the DB2 SQL Performance Analyzer stored procedures ....... 27
    Worksheets: Gathering required data set names .............. 28
  Worksheets: Gathering parameter values for Tools Customizer ........ 29
  Starting and preparing Tools Customizer for use .................. 60
    Starting Tools Customizer ................................ 60
    Modifying Tools Customizer user settings ..................... 61
    Hiding and displaying panel text ................................ 64
  Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool .......... 65
    Roadmap: Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool for the first time ....... 66
    Roadmap: Customizing a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool from a previous customization .......... 67
    Roadmap: Recustomizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool .................. 68
    Specifying the metadata library for the product to customize .......... 70
    Discovering DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool information automatically ........ 72
Creating and associating DB2 entries ............................................. 74
Defining parameters ........................................................................... 77
Generating customization jobs ............................................................. 83
Submitting customization jobs ............................................................... 84
Browsing parameters ........................................................................... 87
Copying DB2 entries ............................................................................. 87
Removing DB2 entries ......................................................................... 88
Deleting DB2 entries ........................................................................... 89
Displaying customization jobs ............................................................... 89
Maintaining customization jobs ............................................................. 90
Using Tools Customizer in a multiple-LPAR environment ...................... 90
Optional: Creating generic plan tables ............................................... 91
Optional: Interfacing with DB2 Query Monitor ..................................... 93
Optional: Configuring DB2 SQL Performance Analyzer for use with QMF  94
Optional: Interfacing with DB2 Administration Tool ............................. 98
Optional: Set up ANLCSPA for use from Browse or Edit sessions .......... 99
Optional: Enabling the stored procedure customization for DB2 SQL Performance Analyzer .......... 100
  Important requirements for using the stored procedure ...................... 100
  Requirements for allocating the stored procedure load library ..... 100
  Customizing the stored procedure address space startup procedure ...... 100
  Customizing the stored procedure ANLCNTL parameters ............... 101
  Calling the DB2 SQL Performance Analyzer stored procedure .......... 102
  Specifying user parameters to the DB2 SQL Performance Analyzer stored procedure .... 103
Migrating DB2 SQL Performance Analyzer from one DB2 version to another DB2 version ........ 106
  Migrating DB2 SQL Performance Analyzer to DB2 Version 9.1 for z/OS .... 106
  Migrating DB2 SQL Performance Analyzer to DB2 10 ...................... 106

Part 3. Administration ........................................................................... 109

Chapter 3. Running DB2 SQL Performance Analyzer ............................. 111
  Using DB2 SQL Performance Analyzer in batch .................................. 111
    SQL PA Batch JCL ........................................................................ 111
    Generating JCL for batch processing .............................................. 113
    Required ddnames ....................................................................... 113
    Using debug batch JCL ............................................................... 114
    Sort Query Limits Report batch JCL ............................................. 114
  Running DB2 SQL Performance Analyzer using the ISPF interface ........ 114
    Starting SQL PA using ISPF ....................................................... 115
    Navigating the DB2 SQL Performance Analyzer panels ............... 115
    Using scrollable fields on DB2 SQL Performance Analyzer panels .... 116
    DB2 SQL Performance Analyzer primary menu panel .................... 117
    Defining output reports ............................................................ 117
    Modifying DB2 SQL Performance Analyzer parameters ............... 118
    Basic processing options ......................................................... 119
    Comparing old and new plans or packages ................................... 120
    Viewing the last set of generated reports .................................... 120
    ANLCSPA edit macro .................................................................. 121
    What if? analysis ........................................................................ 124
    DB2 SQL Performance Analyzer statistics migration .................... 130
    DB2 SQL Performance Analyzer reports menu .............................. 133
  Using DB2 SQL Performance Analyzer in QMF  ................................ 134
    QMF intercept exit ..................................................................... 134
    DB2 primary options menu ....................................................... 134
    DB2 SQL Performance Analyzer cost summary under QMF ............ 134
    Run queries that exceed limits in QMF ....................................... 135
    QMF batch processing ............................................................. 136
  Using the DB2 SQL Performance Analyzer stored procedure ............ 136
    Accessing the DB2 SQL Performance Analyzer stored procedure .... 137
    Easy Explain report levels ......................................................... 138
    EEPATH table ....................................................................... 140
Chapter 4. DB2 SQL Performance Analyzer run parameters ........................................ 143
Impact on cost estimates .......................................................................................... 143
Parameter file characteristics .................................................................................. 145
Specifying ANLPARM user parameters .................................................................. 145
Specifying threshold control parameters ................................................................ 146
Specifying ANLCTNTL configuration parameters and selecting the target host system .............................................................. 147
Enabling the predicate analysis report ..................................................................... 148
Optimizing the table and SQL report processing ...................................................... 148
The difference between SANLDATA and SANLPARM ............................................. 149
Qualifying unqualified objects in plans and packages .............................................. 149
Qualifying unqualified objects in DBRMs and flat files ............................................ 150

Chapter 5. Easy Explain input specifications ......................................................... 151
Easy Explain process ............................................................................................... 151
Configuring Easy Explain ...................................................................................... 151
Easy Explain plan table ........................................................................................... 152
Easy Explain input command requirements ............................................................. 152
Easy Explain input commands ................................................................................ 153
Input command definitions ...................................................................................... 160
Subparameter definitions ....................................................................................... 163
Specifying language and explaining data from another user .................................... 170
Selected authorization identification ....................................................................... 171
User authorization table ......................................................................................... 172
User defaults data set ............................................................................................. 172
Transferring complex queries to SQL PA ............................................................... 173

Chapter 6. DB2 SQL Performance Analyzer reporting capabilities ...................... 175
DB2 SQL Performance Analyzer cost summary report ............................................ 175
  Resource consumption estimates ......................................................................... 176
  QUNITS values ....................................................................................................... 177
  Charge back cost estimates .................................................................................. 178
  Warnings and notifications .................................................................................... 178
DB2 SQL Performance Analyzer enhanced explain report ....................................... 179
  Table space scans ................................................................................................ 184
  Cluster matching index scans ............................................................................... 186
  Cluster nonmatching index scans ......................................................................... 188
  Random matching index scans ............................................................................ 189
  Random nonmatching index scans ....................................................................... 191
  Multiple index scans ........................................................................................... 192
  Joining tables ....................................................................................................... 194
  Subquery processes .............................................................................................. 198
  Sorting results ...................................................................................................... 201
  Insert, Update, Delete, Select, and Refresh ............................................................ 202
DB2 SQL Performance Analyzer detail trace report ............................................... 206
  Forecasting query processing ............................................................................... 207
  List of the DB2 SQL Performance Analyzer install parameters ............................. 207
  Processing steps .................................................................................................... 211
  I/O estimates ......................................................................................................... 214
  Path lengths and CPU time .................................................................................. 215
  Physical and logical I/O ....................................................................................... 218
  Elapsed wait time ................................................................................................. 219
  Sample DB2 SQL Performance Analyzer detail trace report ............................... 220
QLIMIT (installation limits) report ........................................................................... 223
Predicate Analysis information ............................................................................... 226
Part 4. Troubleshooting .............................................. 245

Chapter 7. DB2 SQL Performance Analyzer troubleshooting .......... 247
Gathering diagnostic information ........................................ 247
Resetting the DB2 SQL Performance Analyzer REGISTRY .............. 247
TSO interfacing notes for installers ....................................... 248
DB2 SQL Performance Analyzer variables in the ISPF user profile ...... 248
Host variable support from any TSO input medium ...................... 248
SQL PA supports upper and lower case .................................... 248
Applying user indexes to tables ......................................... 249

Chapter 8. Tools Customizer troubleshooting .......................... 251
Gathering diagnostic information ........................................... 251
Determining the trace data set name ........................................ 251

Chapter 9. How to look up message explanations ....................... 253

Part 5. Reference .................................................. 255

Chapter 10. Reference information ..................................... 257
Tools Customizer reference ................................................ 257
Tools Customizer terminology and data sets ............................. 257
ANLCNTL configuration parameters ..................................... 262
ANLPARM user parameters ............................................... 267
ANLPARM user parameter settings for QMF ............................. 279
Easy Explain table contents ............................................... 280
The EEEAUTH table .................................................. 280
The EEWORK table .................................................. 280
The EEEDBRM table .................................................. 281
The EEEHINT table .................................................. 282
The EEEP PLAN table ............................................... 282
The EEEP PATH table ............................................... 283
PLAN_TABLE contents ............................................. 284

Chapter 11. Messages and return codes ................................ 289
DB2 SQL Performance Analyzer messages and return codes .......... 289
Return codes ...................................................... 289
Output report messages ............................................... 289
Warnings and alerts ................................................ 327
Notes and recommendations .......................................... 332
Good news and guidelines ........................................... 340
Statistics collector messages .......................................... 345
DB2 SQL Performance Analyzer ISPF panel messages ................ 347
Tools Customizer messages ............................................ 356
Tools Customizer messages ............................................ 357

Notices .......................................................... 415
Trademarks ...................................................... 417
Privacy policy considerations ........................................ 417

Index ............................................................ 419
About this information

IBM® DB2® SQL Performance Analyzer for z/OS® (also referred to as DB2 SQL PA or SQL PA) provides DB2 application programmers and database designers with resource usage information and costs associated with SQL queries without having to run them in DB2.

These topics provide instructions for installing, configuring, and using SQL PA.

These topics are designed to help application programmers and database designers perform the following tasks:

- Customize SQL PA
- Produce and interpret cost reports
- Tune queries to achieve maximum performance
- Use the Easy Explain function to better understand access path selection
- Run SQL PA in different environments
- Diagnose and recover from SQL PA problems

To use these topics, you should have a working knowledge of:

- The OS/390® or z/OS operating system
- ISPF
- SMP/E

Always check the DB2 Tools Product publications page for the most current version of this publication:


How to read syntax diagrams

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

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>> symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol indicates that a syntax diagram is continued from the previous line.
  - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).
- 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 syntax element and is used only for readability.

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

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

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

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, column-name). They represent user-supplied names or values.

- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.

- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.

- Footnotes are shown by a number in parentheses, for example (1).
How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other SQL PA documentation, use one of the following options:

- Use the online reader comment form, which is located at: [www.ibm.com/software/data/rcf/](http://www.ibm.com/software/data/rcf/)

- Send your comments by e-mail to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of SQL PA that you are using, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
Part 1. Product overview

The topics in this section provide you with an overview of DB2 SQL Performance Analyzer.
Chapter 1. DB2 SQL Performance Analyzer overview

IBM DB2 SQL Performance Analyzer for z/OS (also referred to as DB2 SQL PA or SQL PA) is a query analysis tool that you can use to optimize the performance of your SQL statements for DB2.

Topics:
- “What does DB2 SQL Performance Analyzer do?” on page 4
- “DB2 SQL Performance Analyzer features and benefits” on page 4
- “DB2 SQL Performance Analyzer components and facilities” on page 11
- “DB2 SQL Performance Analyzer architecture and process flow” on page 14
- “DB2 SQL Performance Analyzer operating environments” on page 15
- DB2 SQL Performance Analyzer documentation and updates
- Accessibility features

What’s new in DB2 SQL Performance Analyzer

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

SC19-4145-01
- The INCLUDE NULL KEYS option for creating an index has been added. For more information, see INCLUDE NULL KEYS.
- All references to the EEE report have been changed to the plan table report.

SC19-4145-00
- DB2 SQL Performance Analyzer is not supported by DB2 8.
- Functionality to create and drop real and virtual indexes has been added. For more information, see “Creating and dropping indexes” on page 126.
- A checklist and worksheets for planning to customize DB2 SQL Performance Analyzer have been added. For more information, see “Preparing to customize DB2 SQL Performance Analyzer” on page 23.
- A parameter that authorizes the use of the user-level user parameters has been added. For more information, see (USRPARM) Allow users to have a copy of the user parameters.
- The user parameters have been categorized. For more information, see “Worksheets: Gathering parameter values for Tools Customizer” on page 29.
- The following parameters have been removed:
  - MIPRATE
  - NUMZIIP
  - SORTBUF
  - SRMCONS

© Copyright IBM Corp. 2002, 2014
The following customization template names have been changed. For complete descriptions of the tasks, steps, and parameters that are associated with each template, see “Worksheets: Gathering parameter values for Tools Customizer” on page 29.

<table>
<thead>
<tr>
<th>Previous template name</th>
<th>Current template name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANLCRE8D</td>
<td>ANLDCRE8</td>
</tr>
<tr>
<td>ANLCRE8I</td>
<td>ANLICRE8</td>
</tr>
<tr>
<td>ANLCRE8J</td>
<td>ANLJCRE8</td>
</tr>
<tr>
<td>ANLQMFXT</td>
<td>ANLXTQMF</td>
</tr>
<tr>
<td>ANLQMFGV</td>
<td>ANLGVQMF</td>
</tr>
</tbody>
</table>

Support for DB2 objects in UNICODE has been added. For more information, see DB2 objects encoding.

Support for specifying the volume where DB2 SQL Performance Analyzer allocates data sets has been added. For more information, see Customized library volume.

**What does DB2 SQL Performance Analyzer do?**

DB2 SQL Performance Analyzer provides you with an extensive analysis of SQL queries without running them. This analysis helps you to tune your queries to achieve maximum performance.

DB2 SQL Performance Analyzer prevents problems associated with long-running queries by calculating the cost of queries before processing them. DB2 SQL Performance Analyzer tells you how long queries will take before you run them, before resources are consumed, and before the query is terminated by a governor.

DB2 SQL Performance Analyzer determines the cost of running a query under the attach facilities of IMS™, CICS®, batch, TSO, SPUFI, and QMF®. The monetary cost of each query is presented and a cost estimate in familiar units: CPU time, I/O count, elapsed time, and as QUNITS (a single number that represents the overall cost). Data and content are presented in easily readable reports.

Easy Explain provides you with information about how DB2 accesses data for a given SQL statement. Easy Explain can be used in both TSO and batch. Easy Explain compares the costs of old and new plans and provides the ability to re-explain existing plans that are stored in any plan table. Programming language data sets can be scanned directly to extract SQL for analysis.

**DB2 SQL Performance Analyzer features and benefits**

SQL PA includes several performance features that provide for fast and efficient processing without affecting the system.

Topics:

- “Generic plan tables” on page 5
- “Catalog access” on page 5
- “DB2 SQL Performance Analyzer and QMF” on page 5
- “Remote operation through DRDA” on page 6
- “Accelerated table and SQL report processing” on page 6
- “DB2 SQL Performance Analyzer stored procedure capability” on page 6
Generic plan tables

To eliminate the need for each user to have their own authid.PLAN_TABLE for the EXPLAIN output, DB2 SQL Performance Analyzer can use a set of generic plan tables in a specifically designed segmented table space.

Using generic plan tables provides for both immediate and simultaneous access, scratch, and reuse of the generic tables with mass delete (a single I/O to drop all entries). By registering each generic copy in use, you can avoid conflicts during concurrent use of DB2 SQL Performance Analyzer. Concurrent DB2 SQL Performance Analyzer users can process against these generic plan tables without conflict. You can change the settings during configuration. You can also create generic DSN_STATEMNT_TABLEs and DSN_FUNCTION_TABLEs.

If you are not using secondary authorization IDs and generic plan tables, you can use SQL PA under a primary authorization ID, if a valid plan table exists under that authorization ID. DB2 SQL Performance Analyzer automatically detects this situation and switches from the generic to the primary plan table. It is recommended that users in this situation avoid assigning query numbers above 100 000 000 for their own Explains, because DB2 SQL Performance Analyzer deletes all queries above this number upon invocation.

To provide direct selection of specific PLAN_TABLEs, DSN_STATEMNT_TABLEs, and DSN_FUNCTION_TABLEs, DB2 SQL Performance Analyzer provides the USEPLAN and SETPLAN parameters.

Catalog access

DB2 SQL Performance Analyzer accesses the catalog for information about the table spaces, tables, indexes, and relationships that are determined by each access path, and retrieves critical size and cardinality data that is used as input to the cost assessment process.

DB2 SQL Performance Analyzer accesses the catalog only once for each object sought, even if many queries use the same tables or indexes in their access path. DB2 SQL Performance Analyzer stores the details for hundreds of each object type in memory, and checks this information cache before attempting to access the catalog again. Catalog access is available for the SYSTABLESPACE, SYSTABLES, SYSINDEXES, SYSKEYS, SYSCOLUMNS, SYSSYNONYMS, and SYSRoutines tables.

DB2 SQL Performance Analyzer and QMF

DB2 SQL Performance Analyzer interfaces with QMF by using the QMF governor exit, and contains code that is specifically designed for optimum performance with minimal elapsed time. For example, virtually all I/Os to disk are eliminated and are replaced by memory resident arrays.

The PL/I environment is essential to running the DB2 SQL Performance Analyzer cost components. The PL/I environment is set up once upon entry into QMF, and remains ready for processing each query. When QMF terminates, the PL/I environment is disabled.
All nonessential code has been removed from the load module ANLQMF. Also, the source code for the exit interface is provided so that you can define the DB2 SQL Performance Analyzer parameters that describe the environment directly into the exit. By doing so, you can eliminate the processor usage that is required to read the parameters from a file for each run.

**Remote operation through DRDA**

DB2 SQL Performance Analyzer can run a remote copy of DB2 SQL Performance Analyzer on another DB2 subsystem, on the same z/OS server, or on another z/OS server by using the DRDA® protocol and the SQL CONNECT statement.

The remote locations must contain fully installed versions of DB2 SQL Performance Analyzer, including a REGISTRY and PLAN_TABLE table. Then, users can initiate DB2 SQL Performance Analyzer on any system that contains a valid copy of the product and run their SQL by using any other valid copy and DB2 subsystem through DRDA Connect facilities. Results are returned back to the local system for presentation, storing report files, and other tasks. This feature provides the ability to quickly diagnose production problems.

This feature operates in TSO and batch modes only.

**Accelerated table and SQL report processing**

The FASTTBL parameter provides you with the ability to bypass the Easy Explain processing so that you can produce table reports faster.

The FASTTBL parameter is one of the system parameters. Depending on the value of the parameter certain information might not be shown. For example, information about shared or exclusive locks is not shown when FASTTBL=YES.

**DB2 SQL Performance Analyzer stored procedure capability**

DB2 SQL Performance Analyzer can operate as a stored procedure, which enables users from anywhere in the network to call DB2 SQL Performance Analyzer for a cost estimate in real time, from DB2 applications, IMS, CICS, remote workstations, PCs, or any other compatible connection.

Users receive a quick and comprehensive view of how long their queries are likely to run, along with any warning flags about exceeding the limits that are defined in the configuration parameters.

The IBM Resource Limit Facility (RLF) governs the processor time that is spent for dynamic queries only, and only within the DB2 address spaces. If the limit is exceeded, the query ends while in process. However, with the DB2 SQL Performance Analyzer stored procedure, you can develop applications with a built-in governor for queries that can preempt any RLF cancellations. This feature provides you with control and flexibility to handle larger requests.

Whether the query is running under IMS, CICS, or is from a remote client workstation, the following call to the DB2 SQL Performance Analyzer stored procedure is required:

```sql
EXEC SQL CALL ANLPRCR (<varname>)
```

The parameter list includes the SQL statement to be analyzed, its length, any DB2 SQL Performance Analyzer user parameters that you want to work with, and
several output host variables that are populated with DB2 SQL Performance Analyzer cost estimates and warning flags.

DB2 SQL Performance Analyzer provides a stored procedure for use with WLM-controlled address spaces. This stored procedure is named ANLPRCR and uses the Recoverable Resource Manager Services Attach Facility (RRSAF).

A PL/I sample program and a COBOL sample program are provided for the stored procedure environment. Use these sample programs to learn about and design programs that use the WLM stored procedure. Both the source code and the executable modules of the PL/I sample programs are included.

SQL Advisor
The SQL Advisor finds and presents opportunities for fine-tuning performance.

It uses a comprehensive set of warnings, alerts, and recommendations to convey this information. The SQL Advisor can also issue guidelines, notes, and other informational messages to assist you in understanding the logic behind certain design decisions and tuning choices. All advice is integrated into the Explain and Detail Trace reports. You control how much information is generated.

Exclude WHERE CURRENT OF statements from Query Limit reports
The WHERE CURRENT OF clause can cause values that exceed limits set for acceptable query statements. If you are familiar with all of your WHERE CURRENT OF clauses, you can exclude them from your Query Limit reports.

The DSPWCOC parameter provides you the ability to exclude the WHERE CURRENT OF statements from the Query Limits report. Set this parameter to NO to have WHERE CURRENT OF statements excluded from the Query Limits report.

Actual query cost
The total impact of SQL queries that are processed by DB2 is difficult to assess. One reason is because obtaining and recording all of the cost data in one place for a specific query is difficult.

For example, DB2 records some of the processor time that is used by a specific query in the Type 101 Accounting SMF record. However, this time does not represent all of the processor time that was used by the query.

The Sequential and List Prefetch, Asynchronous Writes, Thread Management, part of Lock Management, and other processes are performed by DB2 for you, and charged to one of its own address spaces. Consider the DB2 processor time distribution, as shown in the following figure.
Processor time distribution has the following four levels.

**LEVEL 1**
This level includes the entire ASCB CPU times that are recorded for a typical transaction, including the IRLM and Application A/S (CICS, IMS). This level represents the full cost of a transaction in processor consumption terms.

**LEVEL 2**
This level includes that portion of the CPU times that are reported by DB2 as CPU times (TCB, SRB, and Total) in SMF 101 of the individual user.

**LEVEL 3**
This level includes the CPU times that DB2 does not report for each user. The CPU times are reported for the entire system (ASCB CPU Time) in SMF 100.

**LEVEL 4**
This level includes the application init/term/overheads and program processing that is not recorded by DB2.

As the previous figure shows, assembling all the components of DB2 processor time is not a trivial matter. Looking at the processor time in the SMF Type 101 accounting record yields only part of the processor time that is used by the query. The remainder of processor time is included in the SMF Type 100 statistics record, along with the usage for many other queries, recorded at periodic time intervals. Some components of DB2 processor times are not recorded at all. Although RMF™ can provide the total resource consumption by address space, only DB2 SQL Performance Analyzer can correlate these costs to the specific users who caused them.

Consider the following breakdown of processor time.

- **SRB (always run before any TCB):**
  1. Database writes (async write)
  2. Sequential and list prefetch
3. Part of thread creation

**TCB (in priority sequence):**
1. VSAM open and close
2. Space management
3. BSDS activity
4. Statistics recording

*Within the DB2 SYSTEM SERVICES:*

**SRB work:**
1. Checkpoint logging
2. Physical logging

**TCB work:**
1. Archiving
2. Space management

*Within the DB2 IRLM:*

**TCB Work:**
1. Deadlock detection

*Within the USER (Application) address space:*

**TCB activity:**
1. SQL processing
2. Sync reads
3. Sync writes
4. Lock requests or acquire
5. Lock release
6. Logical logging

Much of the processing that happens outside the scope of the address space of the user is absorbed by DB2. The true cost of each DB2 query might not be adequately represented by the SMF 101 accounting record or by the DB2PM accounting report (or its equivalents). Because DB2 absorbs some of the query costs, some users consider the processor usage of DB2 to be high. The bulk of this processor usage is merely DB2 working on behalf of its users and absorbing the costs internally. Even I/O that is performed by the Media Manager under Database Services A/S is not recorded reliably by SMF within the records (including Job and Step termination) of the user.

In a DB2 SQL Performance Analyzer cost summary, all of the processor time factors are considered. The fully loaded costs of each query are represented in a forecast of resource consumption that is based on each particular form of query.

DB2 SQL Performance Analyzer can split the cost analysis into incremental components, so that you can see exactly where each query is spending most of its time. DB2 SQL Performance Analyzer does analysis without having to run the query, which is valuable for long-running queries against large databases.

Along with the processor time for asynchronous I/O, DB2 SQL Performance Analyzer forecasts the actual counts of sequential and list prefetch, and synchronous read I/O counts, all of which are sometimes difficult to predict. Even more so, the asynchronous writes of those pages that are updated by inserts, updates, and deletes, and the DB2 logging activity that is associated with those writes, are almost never shown in the Type 101 Accounting record. The writes are
not shown because they occur after the query terminates. DB2 SQL Performance Analyzer can correlate them directly to the query that caused them. The writes are used to determine a realistic response time that includes the processor usage of the delivery vehicle, be that TSO SPUFI, or IMS Wait for Input. Consider the elapsed time distributions that are shown in the following figure.

![Figure 2. Elapsed time distribution](image)

The entire elapsed time of the transaction includes some portions that are not captured in DB2 class 1 elapsed time, such as create and terminate thread, plus any DBMS processor usage (CICS/IMS init/term/scheduling).

Class 2 elapsed time is a subset of class 1 and represents the time spent in DB2 asynchronous and synchronous only. Class 3 elapsed time includes the synchronous I/O and lock wait times, and is a subset of class 2. It also includes the asynchronous I/O wait times, and Service Task (SRB) and Archive Log waits.

The last component of elapsed time is application time, which is partially captured in DB2 class 1 for SQL-related activities, and partially not captured for subsequent processing of the rows after fetch.

DB2 SQL Performance Analyzer provides an end-to-end view of elapsed time for each of your queries. It also highlights the work done by the system, after your query ends, in support of insert, update, and delete pages that are logged and rewritten to DASD on your behalf. DB2 SQL Performance Analyzer can provide a more resource-intensive view of how your SQL is processed, and a direct comparison back to SMF accounting records validates part of the cost that is presented by DB2 SQL Performance Analyzer.
DB2 SQL Performance Analyzer components and facilities

DB2 SQL Performance Analyzer is comprised of several main components. DB2 SQL Performance Analyzer uses the DB2 call attachment facility (CAF) or the recovery resources manager attach facility (RRSAF) to interface directly with DB2.

Topics:
- "Parser and catalog interfaces"
- "Resource estimator"
- "QMF Intercept" on page 12
- "CAF and QMF features that enhance SQL PA" on page 12
- "Easy Explain" on page 13
- "Tools Customizer overview" on page 14

Parser and catalog interfaces
The parser and the catalog interfaces find important SQL and store it in a list.

SQL parser routine
The SQL parser routine finds the executable SQL in any sequential file, PDS member, or DBRM module (all DBRMs are supported). The SQL parser extracts those statements and assigns Explain query numbers to them. The following numbering convention is used:
- Sequential SQL begins with 100 000 001.
- SQL statements that are extracted from DBRM modules are identified by the number 100 000 000 + the DBRM statement number.

The role of the SQL parser is to parse and store information about each query, such as the type and number of predicates, sort clauses, columns, and correlation names. This information is used later during the costing phase. The Explain-ready SQL statements are then passed to DB2 for evaluation by use of the EXPLAIN statement.

Catalog Interface
The Catalog Interface component of DB2 SQL Performance Analyzer looks up the access paths that are chosen for each query, the tables and indexes involved, referential integrity relationships, space information, and other Explain data, and then disposes of the generic plan table entries with a mass delete.

DB2 SQL Performance Analyzer accesses the catalog for information about the table spaces, tables, indexes, keys, and relationships discovered by each access path, and retrieves critical size and cardinality data that are essential to the cost assessment process.

DB2 SQL Performance Analyzer accesses the catalog only once for each object sought, even if many queries are using the same tables or indexes in their access path. DB2 SQL Performance Analyzer stores the details about each object type in memory, and checks this information cache before attempting to access the catalog again. Catalog access is mainly confined to the SYSTABLESPACE, SYSTABLES, SYSINDEXES, SYSKEYS, SYSSRELS, SYSCOLUMNS, SYSSYNONYMS, and SYSROUTINES tables.

Resource estimator
The resource estimator component of DB2 SQL Performance Analyzer takes the parser and catalog information and determines the cost of each SQL statement.
The resource estimator determines costs by using its knowledge of the algorithms for the DB2 optimizer, and of the hundreds of benchmarked path lengths that represent the incremental costs of DB2 operations.

The costing module of the resource estimator determines a fully loaded cost for each query by using all of the intelligence data that is provided by the parser, the catalog sizing information in the catalog, and the evaluation techniques of the optimizer.

For example, the resource estimator calculates the filtering effects of each predicate in the WHERE clause to determine a final estimate of the number of rows retrieved. It also determines the portion of the data table and indexes that were accessed to provide these rows. In this processing, DB2 SQL Performance Analyzer mimics the algorithms of the optimizer and uses its costing methods to arrive at the estimate that is provided by DB2 SQL Performance Analyzer. The resource estimator is also sensitive to user and installation parameters, and can hypothesize and predict virtually any DB2 operational situation or environment that you can envision. DB2 SQL Performance Analyzer can also use the internal estimates of the optimizer for path length by using data from the DSN_STATEMNT_TABLE table.

**QMF Intercept**

The QMF Intercept program, ANLQMF, is a condensed version of ANLSQLPA.

The QMF Intercept program manages single SQL statements under QMF and has minimal reporting requirements.

DB2 SQL Performance Analyzer interfaces with QMF by using the QMF governor exit and contains customized code that is designed for optimum performance with minimal elapsed time. Among the performance features is the elimination of virtually all I/Os to disk by replacing them with memory resident cache.

A PL/I environment is required to run the DB2 SQL Performance Analyzer cost components. That environment is set up once, upon entry into QMF. It remains ready for the processing of each query until QMF ends, which eliminates the constant processor usage of re-initializing the PL/I environment.

All nonessential code was removed from the load module, and the source code for the exit interface is provided so you can hard code the parameters that describe the environment into the exit. Hard coding the parameters avoids reading them in for each operation.

Generally, the QMF Intercept program is installed with options that show costs for only those SQL statements that exceed the guidelines that are defined in the configuration parameters. DB2 SQL Performance Analyzer runs in the background and only displays information if your queries are resource-intensive.

**CAF and QMF features that enhance SQL PA**

You can use the call attachment facility (CAF) and the QMF governor exit to enhance how DB2 SQL Performance Analyzer functions.

**CAF**  
DB2 SQL Performance Analyzer uses the CAF to communicate with DB2 instead of using IKJEFT01 because CAF uses fewer resources.

DB2 SQL Performance Analyzer provides its own set of generic plan tables for collecting each user's transient Explain information, and provides a REGISTRY table to control contention and provide maximum concurrency.
These tables are built under DB2 on the systems that DB2 SQL Performance Analyzer is installed on. The secondary authorization ID exit programs that are provided with DB2 (DSN3@ATH and DSN3@SGN) are used to introduce a set of DB2 SQL Performance Analyzer secondary authorization IDs that own the set of DB2 SQL Performance Analyzer generic plan tables.

**QMF governor exit**

The QMF governor exit that is provided with DB2 can intercept all QMF queries and perform a cost analysis on them before the database is accessed. The DB2 SQL Performance Analyzer stored procedure is loaded into the DSNSPAS or WLMSPAS address spaces for access from anywhere in the DB2 network.

DB2 SQL Performance Analyzer supplements the normal facilities of TSO ISPF by adding to the panel and by adding CLIST and message libraries that are commonly allocated by ISPF to plug DB2 SQL Performance Analyzer into the framework of daily operations.

**Easy Explain**

Easy Explain helps the DB2 application designer and programmer understand the access path that was selected by DB2 for a given SQL statement.

Easy Explain uses information that is produced by the SQL EXPLAIN statement and additional relevant DB2 catalog data.

Easy Explain can run in stand-alone mode or can be integrated with DB2 SQL Performance Analyzer. Easy Explain has been fully integrated into the TSO ISPF interface of DB2 SQL Performance Analyzer and is run with each SQL EXPLAIN statement that is performed. Easy Explain can also be run as a batch job. When Easy Explain is integrated, you can extract plans and packages from the DB2 catalog and present them to DB2 SQL Performance Analyzer for analysis. You can also store and compare before-and-after access plans and their costs.

Information that is produced by the SQL EXPLAIN function is useful in application design and tuning. However, the SQL EXPLAIN function information is not sufficient to determine and understand the access path that is chosen by DB2 and documented by the EXPLAIN function.

Typically, the application designer and programmer must extract DB2 catalog information to understand and validate the chosen access path.

The main purpose of Easy Explain is to combine information that is returned by the SQL EXPLAIN statement and information that is extracted from the DB2 catalog and to document that data in an understandable way.

Easy Explain accepts a query number of a previous EXPLAIN operation, the name of a plan that was bound with the EXPLAIN option, a QMF query, or any Data Manipulation Language (DML) statement. For a QMF query or a DML statement, Easy Explain explains the statement and provides an explanation of the selected access path. If the plan or package was not bound with the EXPLAIN option, Easy Explain performs a dynamic EXPLAIN of the DML SQL statements in the plan or package.

Using current EXPLAIN statement data as input to the DB2 SQL Performance Analyzer cost analysis provides you the opportunity to ensure that the DB2 catalog
represents an accurate view of this application environment. For an application in
the early stages of development, that view could be the projected production
environment or it could represent a new indexing option that is being considered
to boost the performance of an existing production application.

**Tools Customizer overview**

IBM Tools Customizer for z/OS (also referred to as Tools Customizer) standardizes
many of the customization processes that are required to customize IBM Tools that
run on z/OS. Tools Customizer is a component of IBM Tools Base for z/OS.

Tools Customizer provides a consistent ISPF interface to ensure that the
customization process is the same for all IBM Tools products and solution pack
components. It also provides the ability to “discover” parameter values from
products or solution pack components that you previously customized manually or
by using Tools Customizer.

**Features and benefits**

Tools Customizer provides the following features:

- A single, consistent ISPF interface ensures that the customization process is the
  same for all IBM Tools products and solution pack components.
- A Discover EXEC discovers values for common product, LPAR, and DB2
  parameters from a product or solution pack component that you previously
  customized manually or by using Tools Customizer. Each IBM Tools product and
  solution pack component has a unique Discover EXEC. The discovered
  parameters are stored in the data store. If the product or solution pack
  component that you want to customize exists in the Tools Customizer data store,
  Tools Customizer issues a warning before it overwrites existing values. Use the
  Discover EXEC by issuing the DISCOVER command on the Customizer
  Workplace panel.
- The data store retains discovered and manually specified parameter values.
  Because the parameter information is persistently stored, you have to manually
  specify or discover parameter values only once. Tools Customizer uses these
  parameter values where they are applicable.
- A metadata repository contains the members that define the following
  customization attributes for products and solution pack components:
  - Parameters, tasks, and steps for the product or solution pack component to be
    customized. Some product or solution pack parameters, tasks, and steps are
    required.
  - LPAR parameters for the local LPAR. All of the LPAR parameters are
    required.
  - DB2 parameters for the DB2 subsystem, DB2 group attach name, or DB2 data
    sharing member on which you will customize the product or solution pack
    component. All of the DB2 parameters are required.
- Default values are provided for product parameters and solution pack
  component parameters, LPAR parameters, and DB2 parameters. The default
  values show examples of how to complete fields.

**DB2 SQL Performance Analyzer architecture and process flow**

DB2 SQL Performance Analyzer processes information from various input sources
based on user and system parameters that you specify and then produces a series
of reports that you use in your analysis.
DB2 SQL Performance Analyzer first reads and sets user and system-related parameters. After the parameters are read, DB2 SQL Performance Analyzer reads SQL statements. The source of the SQL statements can be any of the supported input sources, such as a single DBRM or a whole DBRM library, a PLAN or PACKAGE from the system catalog, SQL that is contained in a sequential or partitioned data set, or SQL that is embedded in source code. For each of the SQL statements, DB2 SQL Performance Analyzer determines if a plan exists. If a plan does not exist, DB2 SQL Performance Analyzer performs a DB2 PREPARE and EXPLAIN to create the necessary plan information. DB2 SQL Performance Analyzer then fetches the rows from various EXPLAIN tables for analysis, fetches the system catalog statistics that are associated with DB2 objects that are being analyzed, analyzes the information collected, and produces a number of reports.

The following figure shows the basic process flow for DB2 SQL Performance Analyzer.

**Figure 3. The DB2 SQL Performance Analyzer process**

---

**DB2 SQL Performance Analyzer operating environments**

DB2 SQL Performance Analyzer operates in four primary environments: batch, TSO, QMF, and as a stored procedure.
From the batch or TSO environment, DB2 SQL Performance Analyzer can evaluate queries that are targeted to run from any attach facility. You determine how your application’s queries attach to DB2 by setting a user parameter that you supply to DB2 SQL Performance Analyzer. To evaluate how your application runs under the many versions of IMS or CICS, run DB2 SQL Performance Analyzer in batch mode to produce a series of reports that detail the forecasted performance of each query.

The same facilities that are available to users of DB2 SQL Performance Analyzer in batch mode are provided online under TSO. When you are designing an application, you can use DB2 SQL Performance Analyzer under TSO iteratively to refine the design of that application. With DB2 SQL Performance Analyzer under TSO, you also can target a few specific SQL statements from a file, and do a detailed cost analysis on these. By using this method, you can examine the effects of various options in the design process. Similarly, you can examine large files of SQL statements or scan DBRM modules.

DB2 SQL Performance Analyzer under TSO and batch provides identical report formats. Therefore, if the system is busy, consider submitting a batch job; if you are fine-tuning an application, do it online in real time by using DB2 SQL Performance Analyzer under TSO.

In the QMF environment, DB2 SQL Performance Analyzer is designed to intercept ad hoc queries and notify you of their cost before processing. Its primary purpose is to identify long-running queries, and give you the opportunity to cancel them before they run.

You can cancel queries that exceed the guidelines for processor use, I/O use, and elapsed time. You can configure queries to be canceled automatically or at your discretion. You can also redirect long-running queries to QMF batch where they can run to completion. When queries are unrestricted by the governor process, you can obtain the results set that you want.

DB2 SQL Performance Analyzer does not intercept IMS or CICS transactions in flight because these systems do not have a common facility to present the cost and solicit a response to continue or cancel. However, IMS and CICS can call the WLM stored procedure (ANLPCCR), which provides a convenient way to govern virtually all DB2 applications, both dynamic and static SQL. You can call DB2 SQL Performance Analyzer as a stored procedure, and pass the SQL statement for evaluation. DB2 SQL Performance Analyzer returns the cost values, warning flags, and other information in real time, so that the application can decide whether to proceed with the query or try an alternative query.

DB2 SQL Performance Analyzer can parse and extract the relevant SQL statements from the DBRM modules that are created by the DB2 precompiler. A DB2 SQL Performance Analyzer cost analysis is then performed (either from TSO or batch mode), and even IMS- or CICS-related costs are included in the overall view of each query. For example, you can use DB2 SQL Performance Analyzer to evaluate an entire application system that is running on IMS to determine which SQL statements are likely to run the longest. You can then fine-tune those statements to enhance their performance. All valid methods of attaching to DB2 are supported by DB2 SQL Performance Analyzer, along with their individual attendant processor usage. The processor usage, including the path lengths of attaching to DB2, is available through private benchmarks.
Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

Note to writers:
This topic is a common topic that gets pulled into your output automatically. Don’t change the contents of this topic or the placement of this topic in your DITA map.

To find service updates and support information, see the following website:

http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_tools_for_z~OS

Product documentation and updates

DB2 Tools information is available at multiple places on the web. You can receive updates to DB2 Tools information automatically by registering with the IBM My Notifications service.

Note to writers:
This topic is a common topic that gets pulled into your output automatically. Don’t change the contents of this topic or the placement of this topic in your DITA map.

Information on the web

The DB2 Tools Product Documentation web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following web page:

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

You can also access documentation for many DB2 Tools from IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter

Search for a specific DB2 Tool product or browse the Information Management > DB2 for z/OS family.

IBM Redbooks® publications that cover DB2 Tools are available from the following web page:

http://www.redbooks.ibm.com

The Data Management Tools Solutions website shows how IBM solutions can help IT organizations maximize their investment in DB2 databases while staying ahead of today's top data management challenges:

Receiving documentation updates automatically

To automatically receive emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:
1. Go to http://www.ibm.com/support/mysupport
2. Enter your IBM ID and password, or create one by clicking register now.
3. When the My Notifications page is displayed, click Subscribe to select those products that you want to receive information updates about. The DB2 Tools option is located under Software > Information Management.
4. Click Continue to specify the types of updates that you want to receive.
5. Click Submit to save your profile.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM product documentation, use one of the following options:
- Use the online reader comment form, which is located at [http://www.ibm.com/software/data/rcf/](http://www.ibm.com/software/data/rcf/)
- Send your comments by email to comments@us.ibm.com. Include the name of the book, the part number of the book, the version of the product that you are using, and, if applicable, the specific location of the text you are commenting on, for example, a page number or table number.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

Note to writers:
This topic is a common topic that gets pulled into your output automatically. Don't change the contents of this topic or the placement of this topic in your DITA map.

The major accessibility features in this product enable users to perform the following activities:
- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
  - z/OS ISPF User’s Guide, Volume 1
  - z/OS TSO/E Primer
  - z/OS TSO/E User’s Guide
These guides describe how to use the ISPF interface, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.
Part 2. Customization

The topics in this section provide you with information about customizing DB2 SQL Performance Analyzer.
Chapter 2. Customizing DB2 SQL Performance Analyzer

After you install DB2 SQL Performance Analyzer by following the installation instructions in the Program Directory, you must run Tools Customizer to specify the variables for each DB2 subsystem and to customize the configuration and user parameters.

To complete the customization process by using Tools Customizer, see “Preparing to customize DB2 SQL Performance Analyzer.”

Preparing to customize DB2 SQL Performance Analyzer

Before you start to customize DB2 SQL Performance Analyzer, determine all of the customization values that you need to specify during the customization process, and familiarize yourself with all of the customization tasks.

The following checklist lists and describes each significant customization step. Use this checklist to guide you through the entire customization process.

Tip: Print the following checklist and the data set names and parameter values worksheets. Use the worksheets to record your values, and refer to them during the customization process.

<table>
<thead>
<tr>
<th>Task</th>
<th>Link to detailed instructions</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools Customizer basics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior to beginning the customization process, familiarize yourself with Tools Customizer terminology and data sets, and other basic information about Tools Customizer.</td>
<td>“Tools Customizer terminology and data sets” on page 257</td>
<td></td>
</tr>
<tr>
<td>Software requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verify that your environment meets the minimum software requirements.</td>
<td>“Verify that your environment meets software requirements” on page 25</td>
<td></td>
</tr>
<tr>
<td>• To install and use DB2 SQL Performance Analyzer, your environment must be running a supported version of the z/OS operating system and of DB2 for z/OS.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• To use the QMF governor, you must be using a supported version of QMF.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• To interface with DB2 Administration Tool for z/OS, DB2 Path Checker for z/OS, and DB2 Query Monitor for z/OS, you must be using supported versions of these products.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMP/E installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verify that DB2 SQL Performance Analyzer has been installed correctly. DB2 SQL Performance Analyzer is installed by using standard SMP/E processing.</td>
<td>“Verify that DB2 SQL Performance Analyzer has been installed successfully” on page 26</td>
<td></td>
</tr>
<tr>
<td>Verify that Tools Customizer for z/OS has been installed correctly. Tools Customizer for z/OS is installed by using standard SMP/E processing.</td>
<td>“Verify that Tools Customizer has been installed successfully” on page 26</td>
<td></td>
</tr>
<tr>
<td>DB2 components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Link to detailed instructions</td>
<td>Status</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Verify that the DSNTIAD sample program, which is used to run SQL statements, is available.</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td><strong>Plan table configurations for DB2 EXPLAIN output</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understand the differences between generic, alternative, or unique EXPLAIN plan tables.</td>
<td>“Plan table configurations for DB2 EXPLAIN output” on page 26</td>
<td></td>
</tr>
<tr>
<td><strong>DB2 SQL Performance Analyzer stored procedure requirements</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensure that the requirements for using the stored procedures are met.</td>
<td>“Requirements for using the DB2 SQL Performance Analyzer stored procedures” on page 27</td>
<td></td>
</tr>
<tr>
<td><strong>Gather data set names</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>During the customization process, you must specify names for the following types of data sets:</td>
<td>“Worksheets: Gathering required data set names” on page 28</td>
<td></td>
</tr>
<tr>
<td>• Tools Customizer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• DB2 SQL Performance Analyzer</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Gather parameter values</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>During the customization process, you must specify parameter values for DB2 SQL Performance Analyzer, for DB2, and for your LPAR.</td>
<td>“Worksheets: Gathering parameter values for Tools Customizer” on page 29</td>
<td></td>
</tr>
<tr>
<td><strong>Customize DB2 SQL Performance Analyzer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel.</td>
<td>“Starting Tools Customizer” on page 60</td>
<td></td>
</tr>
<tr>
<td>Set up Tools Customizer user settings. If you are running Tools Customizer for the first time, you must modify several user settings to suit your environment. Otherwise, if the user settings that you have already established are still appropriate, skip this step.</td>
<td>“Modifying Tools Customizer user settings” on page 61</td>
<td></td>
</tr>
<tr>
<td>Complete the steps in the appropriate customization roadmap based on the type of customization that you are performing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Customizing DB2 SQL Performance Analyzer for the first time</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Follow this roadmap if you do not have a customized version of DB2 SQL Performance Analyzer and you need to customize it for the first time.</td>
<td>“Roadmap: Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool for the first time” on page 66</td>
<td></td>
</tr>
<tr>
<td><strong>Customizing a different version of DB2 SQL Performance Analyzer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Follow this roadmap if you have already customized a version of DB2 SQL Performance Analyzer and you want to use the same parameter values to customize a different version.</td>
<td>“Roadmap: Customizing a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool from a previous customization” on page 67</td>
<td></td>
</tr>
<tr>
<td><strong>Recustomizing DB2 SQL Performance Analyzer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Follow this roadmap if you have a customized version of DB2 SQL Performance Analyzer but you want to change one or more parameter values.</td>
<td>“Roadmap: Recustomizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool” on page 68</td>
<td></td>
</tr>
</tbody>
</table>
Some customization options require you to manually complete additional tasks after you have used Tools Customizer. If you generated jobs in Tools Customizer that correspond to the following customization options, complete the additional tasks before you submit the jobs. In some cases, an optional task can be completed either by using Tools Customizer or by manually completing tasks without using Tools Customizer.

### Optional: Creating generic plan tables
Create secondary authorization IDs that will act as the owners of the generic DB2 SQL Performance Analyzer plan tables.

<table>
<thead>
<tr>
<th>Task</th>
<th>Link to detailed instructions</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optional: Creating generic plan tables</strong></td>
<td>“Optional: Creating generic plan tables” on page 91</td>
<td></td>
</tr>
</tbody>
</table>

### Optional: Interfacing with DB2 Query Monitor
Enable DB2 SQL Performance Analyzer to interface with DB2 Query Monitor.

<table>
<thead>
<tr>
<th>Task</th>
<th>Link to detailed instructions</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optional: Interfacing with DB2 Query Monitor</strong></td>
<td>“Optional: Interfacing with DB2 Query Monitor” on page 93</td>
<td></td>
</tr>
</tbody>
</table>

### Optional: Configuring DB2 SQL Performance Analyzer for use with QMF
Create the interface between DB2 SQL Performance Analyzer and QMF, and customize the interface by modifying the parameters that define the target host system.

<table>
<thead>
<tr>
<th>Task</th>
<th>Link to detailed instructions</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optional: Configuring DB2 SQL Performance Analyzer for use with QMF</strong></td>
<td>“Optional: Configuring DB2 SQL Performance Analyzer for use with QMF” on page 94</td>
<td></td>
</tr>
</tbody>
</table>

### Interfacing with DB2 Administration Tool
Add DB2 SQL Performance Analyzer to the DB2 Admin Launchpad.

<table>
<thead>
<tr>
<th>Task</th>
<th>Link to detailed instructions</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interfacing with DB2 Administration Tool</strong></td>
<td>“Optional: Interfacing with DB2 Administration tool” on page 98</td>
<td></td>
</tr>
</tbody>
</table>

### Optional: Set up ANLCSPA for use from Browse or Edit sessions
Either use Tools Customizer to generate the job or run the ANLCALLC CLIST.

<table>
<thead>
<tr>
<th>Task</th>
<th>Link to detailed instructions</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optional: Set up ANLCSPA for use from Browse or Edit sessions</strong></td>
<td>“Optional: Set up ANLCSPA for use from Browse or Edit sessions” on page 99</td>
<td></td>
</tr>
</tbody>
</table>

### Optional: Enabling the stored procedure customization for DB2 SQL Performance Analyzer
Customizing and calling the stored procedure address space startup procedure.

<table>
<thead>
<tr>
<th>Task</th>
<th>Link to detailed instructions</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optional: Enabling the stored procedure customization for DB2 SQL Performance Analyzer</strong></td>
<td>“Optional: Enabling the stored procedure customization for DB2 SQL Performance Analyzer” on page 100</td>
<td></td>
</tr>
</tbody>
</table>

### Set up your environment prior to customization
Prior to beginning the customization process, ensure that your environment meets all requirements, that you have installed all prerequisite software, and that you have considered how you want to customize optional features.

### Verify that your environment meets software requirements
Ensure that you are using z/OS V1.12 (5694-A01) or later.

Ensure that you are using one of the following supported versions of DB2 for z/OS:
- DB2 V9.1 (5635-DB2)
- DB2 Value Unit Edition V9.1 (5697-P12)
- DB2 V10 (5605-DB2)
- DB2 Value Unit Edition V10.1 (5697-P31)
- DB2 V11 (5615-DB2)
- DB2 Value Unit Edition V11.1 (5697-P43)

**Attention:** Enabling-new-function mode (ENFM) is not supported.

To use the QMF governor, ensure that you are using one of the following supported versions of DB2 QMF:
To use DB2 SQL Performance Analyzer to expand the analysis of the SQL statements that are selected by IBM DB2 Path Checker for z/OS, ensure that you are using one of the following supported versions of DB2 Path Checker:
- DB2 Path Checker V4.1
- DB2 Path Checker V4.2

To add DB2 SQL Performance Analyzer to the DB2 Administration Tool Launchpad, ensure that you are using one of the following supported versions of DB2 Administration Tool for z/OS:
- DB2 Administration Tool V10.1
- DB2 Administration Tool V10.2
- DB2 Administration Tool V11.1

To set up the interface between DB2 SQL Performance Analyzer and DB2 Query Monitor for z/OS, ensure that you are using one of the following supported versions of DB2 Query Monitor:
- DB2 Query Monitor V3.1
- DB2 Query Monitor V3.2

**Verify that DB2 DB2 SQL Performance Analyzer has been installed successfully**

See the program directory for IBM DB2 DB2 SQL Performance Analyzer for z/OS, GI10-8977 for installation instructions.

**Verify that Tools Customizer has been installed successfully**

Tools Customizer for z/OS is a component of IBM Tools Base for z/OS (5655-V93), which is available free of charge. Tools Customizer for z/OS provides a standard approach to customizing IBM DB2 for z/OS Tools.

See the program directory for IBM Tools Base for z/OS, GI10-8819 for installation instructions.

**Plan table configurations for DB2 EXPLAIN output**

You can set up generic, alternative, or unique EXPLAIN plan tables. You can define a unique `authorization_ID.PLAN_TABLE` for each user for DB2 EXPLAIN output. You can optionally choose to create generic plan tables, or you can specify alternative EXPLAIN tables for DB2 EXPLAIN output.

Creating generic plan tables is preferred to specifying alternative EXPLAIN tables. Generic plan tables provide better performance. Generic plan tables also allow important user entries to be overlaid if plan tables under the primary authorization ID of a user are used with DB2 SQL Performance Analyzer. Generic plan tables conserve space and allow DB2 SQL Performance Analyzer to handle ad hoc users who do not have their own EXPLAIN output tables. You can set up an arbitrary number of generic plan tables for general use. These plan tables are shared by all DB2 SQL Performance Analyzer online and batch users.
**Tip:** You might want to run DB2 SQL Performance Analyzer without adding generic secondary authorization IDs when you first start using DB2 SQL Performance Analyzer. In this case, all DB2 SQL Performance Analyzer users must have their own plan tables (authorization_ID.PLAN_TABLE) created before they can use DB2 SQL Performance Analyzer, or they must select one by specifying alternative EXPLAIN tables.

If you cannot set up generic plan tables due to restrictions for creating secondary authorization IDs, you can specify alternative EXPLAIN tables by using the USEPLAN and SETPLAN parameters.

**Requirements for using the DB2 SQL Performance Analyzer stored procedures**

DB2 SQL Performance Analyzer has some important requirements for running DB2 SQL Performance Analyzer as a stored procedure.

The stored procedure facility allows a specialized program or procedure to be called from any DB2 application program, including DB2 clients from PC servers that arrive through DDF, CICS, IMS, or other OS/390 regions. The procedure is called in the same way as a subroutine or subprogram. The procedure accepts host variables as input and returns host variables as output. Rows can also be included as part of the answer set that is returned to the caller.

**Tip:** DB2 SQL Performance Analyzer provides stored procedures that are managed by WLM (ANLPRCR and ANLPRER) and sample access programs for the ANLPRCR and ANLPRER stored procedures in PL/I and the ANLPRCR stored procedure in COBOL.

The DB2 SQL Performance Analyzer stored procedures can run in the WLM address space. z/OS is not aware that the stored procedures are DB2-related programs. Stored procedures must be created and linked with the Language Environment® libraries (LE/370), which is a common set of link-edit and runtime routines that support Assembler, COBOL, PL/I, FORTRAN, and C. These libraries are normally named SYS1.CEE.xxxx.

DB2 SQL Performance Analyzer users can issue a stored procedure call from any z/OS address space to the ANLPRCR stored procedure to determine the cost of the query before they run it. DB2 SQL Performance Analyzer returns the costs of the query, along with warnings and return codes that indicate whether the query runs within the defined limits.

The DB2 SQL Performance Analyzer stored procedure components are distributed in the libraries that are shown in the following table:
Table 1. Stored procedure component library distribution

<table>
<thead>
<tr>
<th>Library</th>
<th>Stored procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td>hlg.SANLOAD</td>
<td>The files here exist in SANLOAD before customization for stored procedures. After customization for stored procedures, the files are copied to SANLODS. Use the files when they are in the SANLODS library.</td>
</tr>
<tr>
<td></td>
<td>• ANLPRCR</td>
</tr>
<tr>
<td></td>
<td>• ANLPRER</td>
</tr>
<tr>
<td></td>
<td>Before the ANLSTPR and ANLSTER programs can be used, you must have completed product customization by using Tools Customizer.</td>
</tr>
<tr>
<td>hlg.SANLDDBRM</td>
<td>ANLPRCR, ANLSTPR, ANLPRER, ANLSTER</td>
</tr>
<tr>
<td>hlg.SANLSAMP</td>
<td>ANLSTPR, ANLSTCR, ANLSTER</td>
</tr>
<tr>
<td>hlg.SANLJCL</td>
<td>The following files will be in SANLJCL after customization is complete:</td>
</tr>
<tr>
<td></td>
<td>• ssidSTER and ssidSTPR (execution JCL)</td>
</tr>
<tr>
<td></td>
<td>• ssidWLM, ssidWLMR, and ssidWLMX, (DB2 set up JCL)</td>
</tr>
</tbody>
</table>

Worksheets: Gathering required data set names

Identify and record the data set names that will be used during the customization process and make sure that requirements for certain data sets are met.

Tip: Print the following worksheets and refer to them during the customization process.

Data set names for Tools Customizer

Identify and record the following Tools Customizer data set names:

<table>
<thead>
<tr>
<th>Data set name</th>
<th>Description</th>
<th>Special requirements</th>
<th>Your data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCCQEXEC</td>
<td>EXEC library for Tools Customizer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SCCQDENU</td>
<td>Metadata library for Tools Customizer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SCCQLOAD</td>
<td>Executable load module library for Tools Customizer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SCCQMENU</td>
<td>ISPF messages for Tools Customizer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SCCQPENU</td>
<td>ISPF panels for Tools Customizer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SCCQSAMP</td>
<td>Sample members for Tools Customizer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SCCQTENU</td>
<td>Table library for Tools Customizer</td>
<td>None.</td>
<td></td>
</tr>
</tbody>
</table>
## Data set names for DB2 SQL Performance Analyzer

Identify and record the following DB2 SQL Performance Analyzer data set names. During the customization process, you will enter the following values on the Tools Customizer panels.

<table>
<thead>
<tr>
<th>Data set name</th>
<th>Description</th>
<th>Special requirements</th>
<th>Your data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANLBASE</td>
<td>Sample library for DB2 SQL Performance Analyzer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SANLCLST</td>
<td>CLIST library for DB2 SQL Performance Analyzer</td>
<td>You must have write access to this data set.</td>
<td></td>
</tr>
<tr>
<td>SANLDBRM</td>
<td>DBRM library for DB2 SQL Performance Analyzer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SANLDENU</td>
<td>Metadata library DB2 SQL Performance Analyzer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SANLOAD</td>
<td>Executable load module library for DB2 SQL Performance Analyzer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SANLMENU</td>
<td>ISPF messages for DB2 SQL Performance Analyzer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SANLPENU</td>
<td>ISPF panels for DB2 SQL Performance Analyzer</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>SANLSAMP</td>
<td>Sample members for DB2 SQL Performance Analyzer</td>
<td>You must have write access to this data set.</td>
<td></td>
</tr>
<tr>
<td>SANLSQL</td>
<td>SQL storage library</td>
<td>You must have write access to this data set.</td>
<td></td>
</tr>
</tbody>
</table>

## Worksheets: Gathering parameter values for Tools Customizer

During the customization process, you provide parameter values for the product that you are customizing, for DB2, and for your LPAR.

Use the worksheets in this topic to record the appropriate parameter settings for your purposes, and then use these worksheets during the customization process. The worksheets are organized based on the order of the customization panels in Tools Customizer.

**Tip:** Print the following worksheets and refer to them during the customization process.

### Settings for Tools Customizer

**Description**

Use the following worksheet to identify and record the values for Tools Customizer settings. During the customization process, you will enter these values on the Tools Customizer Settings panel (CCQPSET).

For more information about the parameters in this section, see "Data sets that Tools Customizer uses during customization” on page 260.
### Product Customization Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sample or default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customization library qualifier</td>
<td>DB2TOOL.PRODUCT.CUST</td>
<td></td>
</tr>
<tr>
<td>Use DB2 group attach</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

### Tools Customizer Library Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sample or default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata library</td>
<td>DB2TOOL.CCQ110.SCCQDENU</td>
<td></td>
</tr>
<tr>
<td>Discover output data set</td>
<td>DB2TOOL.CCQ110.DISCOVER</td>
<td></td>
</tr>
<tr>
<td>Data store data set</td>
<td>DB2TOOL.CCQ110.DATASTOR</td>
<td></td>
</tr>
</tbody>
</table>

### User Job Card Settings for Customization Jobs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sample or default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The job card information to be</td>
<td>The job statement</td>
<td></td>
</tr>
<tr>
<td>inserted into the generated jobs</td>
<td>information from the</td>
<td></td>
</tr>
<tr>
<td>for customizing a product or component</td>
<td>ISPF Batch Selection panel.</td>
<td></td>
</tr>
</tbody>
</table>

### Metadata library for DB2 SQL Performance Analyzer

**Description**

Use the following worksheet to identify and record the value of the metadata library for DB2 SQL Performance Analyzer. During the customization process, you will enter this value on the Specify the Metadata Library panel (CCQPFLQ).
### DB2 SQL Performance Analyzer metadata library

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sample or default value</th>
<th>Discovered?</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata library</td>
<td><code>hlq.SANLDENU</code></td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

### Customization values for the Discover EXEC

Complete this worksheet only if you are recustomizing a product that has previously been customized by using Tools Customizer.

**Description**

Use the following worksheet to identify and record the customization values for the Tools Customizer Discover EXEC. During the customization process, you will enter these values on the Discover Customized Product Information panel (CCQPDISC).

### Discover EXEC for Extracting Information from an Already Customized Product

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sample or default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover EXEC library</td>
<td><code>hlq.SANLCLST</code></td>
<td></td>
</tr>
<tr>
<td>Discover EXEC name</td>
<td><code>ANLCDISC</code></td>
<td></td>
</tr>
<tr>
<td>Discover output data set</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The name of the Discover EXEC.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The fully qualified name of the data set for the output from the product Discover EXEC.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information, see “Data sets that Tools Customizer uses during customization” on page 260.</td>
<td></td>
</tr>
</tbody>
</table>

### Information for Discover EXEC

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sample or default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL PA high-level qualifier from the previous customization of SQL PA</td>
<td><code>SYSX.ANL420</code></td>
<td></td>
</tr>
</tbody>
</table>

For more information, see “Data sets that Tools Customizer uses during customization” on page 260.
**Product to Customize section**

**Description**

The parameters that are listed in the Product to Customize section are read-only. They contain information that was provided on other panels, by Tools Customizer, or by the SQL PA metadata data set.

**Read-only fields in the Product to Customize section on Tools Customizer panels**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Discovered?</th>
<th>Source of this value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product metadata library</td>
<td>No</td>
<td>This value is specified on the Specify the Metadata Library panel (CCQPHLQ).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is the library that you specified on the Specify the Product to Customize panel. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.</td>
</tr>
<tr>
<td>LPAR</td>
<td>No</td>
<td>This value is provided by Tools Customizer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The LPAR field displays the LPAR on which you are customizing DB2 SQL Performance Analyzer.</td>
</tr>
<tr>
<td>Product name</td>
<td>No</td>
<td>This value is provided by the product metadata file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value displays the product that is being customized. In this example, DB2 SQL Performance Analyzer should be displayed in this field. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.</td>
</tr>
<tr>
<td>Version</td>
<td>No</td>
<td>This value is provided by the product metadata file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Version field displays the version, release and maintenance of the product you are customizing in the format Vn.Rn.mm.</td>
</tr>
<tr>
<td>Product customization library</td>
<td>No</td>
<td>This value is derived from the user-specified customization library qualifier on the Tools Customizer Settings panel (CCQPSR).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value displays the name of the data set in which the generated library customization jobs will be stored.</td>
</tr>
</tbody>
</table>

**Required parameters section**

**Description**

The parameters in this task are required for all customizations. During the customization process, you will enter these values on the Product Parameters panel (CCQPPR).

**Parameters in the Required parameters section on the Product Parameters panel (CCQPPR)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLQ of DB2 SQL Performance Analyzer installation</td>
<td>Yes</td>
<td>Yes</td>
<td>SQLPA.ANL420</td>
<td></td>
</tr>
<tr>
<td>The high-level qualifier of the SQL PA libraries.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Task: Create files needed for TSO and batch execution

Description
Sets up the necessary files to be able to invoke SQL PA through TSO or batch.

This task is required.

Jobs generated
The steps in this task generate the following customization jobs:

- **ANLDROab**, where *ab* are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLDROPD template and is in the *job_sequence_numberDROPDB2_entry_ID* member.
- **ANLDCRab**, where *ab* are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLDCRE8 template and is in the *job_sequence_numberDCREDB2_entry_ID* member.
- **ANLPCRab**, where *ab* are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLPCRE8 template and is in the *job_sequence_numberPCREDB2_entry_ID* member.
- **ANLICRab**, where *ab* are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLICRE8 template and is in the *job_sequence_numberICREDB2_entry_ID* member.
- **ANLJCRab**, where *ab* are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLJCRE8 template and is in the *job_sequence_numberJCREDB2_entry_ID* member.
- **ANLBNDab**, where *ab* are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLBNDGR template and is in the *job_sequence_numberBNDGDB2_entry_ID* member.

Required authority
The following list indicates the job and the type of authority that is required to run the job:

- **ANLDROab job**
  The user ID must have DB2 system administrator authority to use the customization ID.

- **ANLDCRab job**
  The user ID must have DB2 system administrator authority to use the customization ID.

- **ANLPCRab job**
  The required authority depends on the following factors:
  - To create the *hlq.SANLPARM* and *hlq.SANLJCL* libraries, the user ID must have the authority to create them by using the customization library high-level qualifier.
  - If the **Copy stored procedure load modules** step in the **Set up stored procedures** task is selected, the user ID must have the authority to create libraries by using the installation high-level qualifier and the customization library high-level qualifier.
  - If the **Create SANLVCLS** step in the **Set up ANLCSPA for use from Browse or Edit sessions** task is selected, the user ID must have the authority to create libraries by using the installation high-level qualifier.

- **ANLICRab job**
  The user ID must have the authority to create members in the installation libraries and the customization libraries.

- **ANLJCRab job**
  The user ID must have the authority to create members in the customization libraries.

- **ANLBNDab job**
  The user ID must have DB2 system administrator authority to use the customization ID.
### Steps and parameters for the Create files needed for TSO and batch execution task

<table>
<thead>
<tr>
<th>Step or parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Drop DB2 objects</strong></td>
<td>No</td>
<td>No</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>Drops DB2 objects from a previous customization or SQL PA version.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Create files, create DB2 objects, and perform binds and grants</strong></td>
<td>Yes</td>
<td>No</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>Sets up the parameter files and the initialization files needed to start DB2 SQL Performance Analyzer. As part of that process, it creates the SANLJCL library if necessary, creates the DB2 objects, and performs the necessary binds and grants for the DB2 SQL Performance Analyzer packages and plans.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Prefix for output reports and work files</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>The default value is your TSO user ID.</td>
<td></td>
</tr>
<tr>
<td>The ANLWRKP user parameter sets the prefix to be used for the output report and work files. The default value is your TSO user ID.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>High-level qualifier of DB2 Path Checker</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>SYSX.DB2PATHC</td>
<td></td>
</tr>
<tr>
<td>The PATHLIB system parameter indicates the high-level qualifier of the libraries in which DB2 Path Checker is installed. If you do not have DB2 Path Checker installed, accept the default value.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Task: Create the generic plan tables

**Description**

Creates the generic plan tables for use by all users.

This task is optional.

**Jobs generated**

This task generates the following jobs:

- ANL3@Aab, where ab are alphanumeric characters assigned by Tools Customizer. This job is based on the ANL3@ATH template and is in the job_sequence_number3@ATDB2_entry_ID member.
- ANL3@Sab, where ab are alphanumeric characters assigned by Tools Customizer. This job is based on the ANL3@SGN template and is in the job_sequence_number3@SGDB2_entry_ID member.

**Required authority**

The user ID that runs the ANL3@Aab and the ANL3@Sab jobs must have DB2 system administrator authority to use the customization ID.
### Steps and parameters for the Create the generic plan tables task

<table>
<thead>
<tr>
<th>Step or parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new DSN3@ATH</td>
<td>Yes</td>
<td>No</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>Creates the modified DSN3@ATH load module.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create new DSN3@SGN</td>
<td>No</td>
<td>No</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>Creates the modified DSN3@SGN load module.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Task: Set up interface to QMF

Complete this worksheet only if you intend to use DB2 SQL Performance Analyzer with DB2 QMF.

**Description**

Sets up the QMF governor exit that is used with DB2 SQL Performance Analyzer.

**Jobs generated**

This task generates the following jobs:

- **ANLGVQab**, where \textit{ab} are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLGVQMF template and is in the \textit{job_sequence_numberGVQMDDB2_entry_ID} member.
- **ANLXTQab**, where \textit{ab} are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLXTQMF template and is in the \textit{job_sequence_numberXTQMDDB2_entry_ID} member.
- **ANLQMFCL**. This job is based on the ANLQMFCL template and is in the \textit{job_sequence_numberQMFCL} member.

**Required authority**

The following list indicates the type of authority that is required to run the job:

- **ANLGVQab job**
  - The user ID must have DB2 system administrator authority to use the customization ID.
- **ANLXTQab job**
  - The user ID must have DB2 system administrator authority to use the customization ID.
- **ANLQMFCL job**
  - The user ID must have the authority to use the installation high-level qualifier.

### Steps and parameters for the Set up interface to QMF task

<table>
<thead>
<tr>
<th>Step or parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new ANLQMF and DSQUEGV1</td>
<td>Yes</td>
<td>No</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>Creates the new ANLQMF load module and creates the modified DSQUEGV1 load module.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QMF user library</td>
<td>Yes</td>
<td>Yes</td>
<td>SYS3.SDSQUSRE</td>
<td></td>
</tr>
<tr>
<td>The data set name of the QMF user library.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QMF load library</td>
<td>Yes</td>
<td>Yes</td>
<td>SYS3.SDSQLOAD</td>
<td></td>
</tr>
<tr>
<td>The data set name of the QMF load library.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Steps and parameters for the Set up interface to QMF task

<table>
<thead>
<tr>
<th>Step or parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create ANLCDSQ1</td>
<td>No</td>
<td>No</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>QMF ISPF panel library</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.SDSQPLBE</td>
<td></td>
</tr>
<tr>
<td>QMF ISPF message library</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.SDSQMLBE</td>
<td></td>
</tr>
<tr>
<td>QMF ISPF skeleton library</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.SDSQSLBE</td>
<td></td>
</tr>
<tr>
<td>QMF REXX execution library</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.SDSQEXCE</td>
<td></td>
</tr>
<tr>
<td>QMF CLIST library</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.SDSQCLTE</td>
<td></td>
</tr>
<tr>
<td>QMF panel library</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.SDSQPNLE</td>
<td></td>
</tr>
<tr>
<td>QMF GDDM map group</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.DSQMAPE</td>
<td></td>
</tr>
<tr>
<td>QMF GDDM chart group</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.DSQCHART</td>
<td></td>
</tr>
<tr>
<td>QMF GDDM data</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.ADMCDATA</td>
<td></td>
</tr>
<tr>
<td>QMF GDDM form definitions</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.ADMGDF</td>
<td></td>
</tr>
<tr>
<td>QMF GDDM symbols</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.ADMSYMBL</td>
<td></td>
</tr>
<tr>
<td>QMF output print file</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.ADMPRINT. REQUEST.QUEUE</td>
<td></td>
</tr>
<tr>
<td>QMF GDDM definitions</td>
<td>No</td>
<td>Yes</td>
<td>SYS3.GDDMDEF1</td>
<td></td>
</tr>
</tbody>
</table>
**Task: Set up interface to Query Monitor**

Complete this worksheet only if you intend to use DB2 SQL Performance Analyzer with DB2 Query Monitor.

**Description**
Creates the CLIST necessary to interface with Query Monitor.

**Jobs generated**
This task generates the ANLQMONC job. This job is based on the ANLQMONC template and is in the `job_sequence_number`QMONC member.

**Required authority**
The user ID that runs the ANLQMONC job must have the authority to use the installation high-level qualifier.

**Steps and parameters for the Set up interface to Query Monitor task**

<table>
<thead>
<tr>
<th>Step or parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create ANLCIQM</td>
<td>Yes</td>
<td>No</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates the ANLCIQM CLIST that is used by Query Monitor to invoke SQL PA.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Task: Set up stored procedures**

Complete this worksheet only if you intend to use the SQL PA stored procedures.

**Description**
Sets up the SQL PA stored procedures.

**Jobs generated**
This task generates the following customization jobs:
- ANLCLODS. This job is based on the ANLCLODS template and is in the `job_sequence_number`CLODS member.
- ANLWLMab, where `ab` are alphanumeric characters assigned by Tools Customizer. This job is based on the ANLWLMSP template and is in the `job_sequence_number`WLMSDB2_entry_ID member.

**Required authority**

**ANLCLODS**
The user ID must have the authority to use the installation high-level qualifier.

**ANLWLMab**
The user ID must have DB2 system administrator authority to use the customization ID.

**Steps and parameters for the Set up stored procedures task**

<table>
<thead>
<tr>
<th>Step or parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy stored procedure load modules</td>
<td>Yes</td>
<td>No</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>Copies the stored procedure load modules from the SANLOAD library to the SANLODS library.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set up the stored procedure using RRSAF</td>
<td>No</td>
<td>No</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>Sets up the stored procedure to use RRSAF.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Task: Set up ANLCSPA for use from Browse or Edit sessions**

Complete this worksheet only if you intend to set up ANLCSPA for use from Browse or Edit sessions.

ANLCSPA is a special edit macro that allows you to selectively pick a portion of the file for analysis, choosing one or several statements.

**Description**

Sets up ANLCSPA for use from Browse or Edit sessions.

**Jobs generated**

This task generates the following customization jobs:

- ANLCVCLS. This job is based on the ANLCVCLS template and is in the `job_sequence_numberCVCLS member`.
- ANLCCSPA. This job is based on the ANLCCSPA template and is in the `job_sequence_numberCCSPA member`.

**Required authority**

The user ID that runs the ANLCVCLS and ANLCCSPA jobs must have the authority to use the installation high-level qualifier and the customization library high-level qualifier.

**Steps and parameters for the Set up ANLCSPA for use from Browse or Edit sessions task**

<table>
<thead>
<tr>
<th>Step</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create SANLVCCLS</td>
<td>No</td>
<td>No</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>Copy ANLCSPA to SYSPROC library</td>
<td>No</td>
<td>No</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>SYSPROC library for ANLCSPA</td>
<td>No</td>
<td>Yes</td>
<td>SYSX.SYSPROC</td>
<td></td>
</tr>
</tbody>
</table>

**Task: Add SQL PA to the DB2 Admin Launchpad**

Complete this worksheet only if you intend to add SQL PA to the DB2 Admin Launchpad.

**Description**

Adds SQL PA to the DB2 Admin Launchpad.

**Jobs generated**

ANLLCHPD. This job is based on the ANLLCHPD template and is in the `job_sequence_numberLCHPD member`.

**Required authority**

The user ID that runs the ANLLCHPD job must have the authority to run the ADBDMTI EXEC in DB2 Admin.
### Steps and parameters for the Add SQL PA to the DB2 Admin Launchpad task

<table>
<thead>
<tr>
<th>Step or parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add SQL PA to Launchpad</td>
<td>Yes</td>
<td>No</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>HLQ of DB2 Admin installation</td>
<td>Yes</td>
<td>Yes</td>
<td>SYSX.DB2ADMIN</td>
<td></td>
</tr>
</tbody>
</table>

### LPAR Parameters section

**Description**
This section contains LPAR parameters. All parameters are required. During the customization process, you will enter these values on the LPAR Parameters panel (CCQOPLPR).

#### ISPF Libraries

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message library</td>
<td>Yes</td>
<td>Yes</td>
<td>ISPF.SISPMLIB</td>
<td></td>
</tr>
<tr>
<td>Panel library</td>
<td>Yes</td>
<td>Yes</td>
<td>ISPF.SISPPLIB</td>
<td></td>
</tr>
<tr>
<td>Skeleton library</td>
<td>Yes</td>
<td>Yes</td>
<td>ISPF.SISPSSLIB</td>
<td></td>
</tr>
<tr>
<td>ISPF table input library</td>
<td>Yes</td>
<td>Yes</td>
<td>ISPF.SISPTLIB</td>
<td></td>
</tr>
<tr>
<td>Link list library</td>
<td>Yes</td>
<td>No</td>
<td>ISPSISPLOAD</td>
<td></td>
</tr>
<tr>
<td>Command procedures library</td>
<td>Yes</td>
<td>Yes</td>
<td>SYS1.PROCLIB</td>
<td></td>
</tr>
</tbody>
</table>
### Language Environment Libraries

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language environment high-level qualifier</td>
<td>Yes</td>
<td>Yes</td>
<td>CEE</td>
<td></td>
</tr>
</tbody>
</table>

The high-level qualifier of the language environment library.

### DB2 Parameters section

**Description**
This section contains DB2 parameters. All parameters are required. During the customization process, you will enter these values on the DB2 Parameters panel (CCQPDB2).

#### General DB2 Information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>Yes</td>
<td>Yes</td>
<td>NFM</td>
<td></td>
</tr>
</tbody>
</table>

The mode in which the DB2 subsystem is running. The following values are valid:
- CM is compatibility mode on all listed DB2 versions except DB2 10.
- CM8 is conversion mode from DB2 V8 on DB2 10.
- CM9 is conversion mode from DB2 Version 9.1 on DB2 10.
- NFM is new-function mode on all listed DB2 versions.

<table>
<thead>
<tr>
<th>Level number</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
</table>

The version, release, and modification level of the DB2 subsystem. The following values are valid:
- 910 is valid only for CM or NFM.
- 101 is valid only for CM8, CM9 or NFM.
- 111 is valid only for CM or NFM.

### DB2 Libraries

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load library</td>
<td>Yes</td>
<td>Yes</td>
<td>DSN.SDSNLOAD</td>
<td></td>
</tr>
</tbody>
</table>

The data set name of the DB2 load library. You can specify multiple values for this parameter.

| Run library | Yes       | Yes         | DSN.RUNLIB.LOAD |            |

The data set name of the DB2 run library. You can specify multiple values for this parameter.
### DB2 Libraries

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit library</td>
<td>Yes</td>
<td>Yes</td>
<td>DSN.SDSNEXIT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample library</td>
<td>Yes</td>
<td>Yes</td>
<td>DSN.SDSNSAMP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro library</td>
<td>Yes</td>
<td>Yes</td>
<td>DSN.SDSNMACS</td>
<td></td>
</tr>
</tbody>
</table>

The data set name of the DB2 exit library. You can specify multiple values for this parameter.

The data set name of the DB2 sample library. You can specify multiple values for this parameter.

The data set name of the DB2 macro library. You can specify multiple values for this parameter.

### DB2 Buffer Pools

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the 4 KB buffer pool</td>
<td>Yes</td>
<td>No</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name of the 8 KB buffer pool</td>
<td>Yes</td>
<td>No</td>
<td>BP8K0</td>
<td></td>
</tr>
</tbody>
</table>

The name of the 4 KB buffer pool to be used for customization. The value must be 8 characters or less.

The name of the 8 KB buffer pool to be used for customization. The value must be 8 characters or less.

### DB2 Utilities

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan name for the DSNTIAD utility</td>
<td>Yes</td>
<td>Yes</td>
<td>DSNTIAD</td>
<td></td>
</tr>
</tbody>
</table>

The plan name for the DSNTIAD utility. The value must be 8 characters or less.

### SQL PA DB2 Objects

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage group name</td>
<td>Yes</td>
<td></td>
<td>SYSDEFLT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database name</td>
<td>Yes</td>
<td>No</td>
<td>ANLDBASE</td>
<td></td>
</tr>
</tbody>
</table>

The name of the storage group that will be used for creating DB2 objects for customization. The value must be 128 characters or less.

The name of the database in which DB2 objects will be created for customization. The value must be 8 characters or less.
### SQL PA DB2 Objects

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table space name</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>ANLSPC08</td>
<td></td>
</tr>
<tr>
<td>The name of the table space in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>which DB2 objects will be created for customization. The value must be 24 characters or less.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SQL PA DB2 General Information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB2 objects encoding</strong></td>
<td>Yes</td>
<td>No</td>
<td>UNICODE</td>
<td></td>
</tr>
<tr>
<td>For DB2 V9.1 and earlier releases, valid values are EBCDIC or UNICODE. For DB2 10 and later releases, any specified value other than UNICODE is ignored, and UNICODE is used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Drop database when dropping DB2 objects</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>To drop the SQL PA database when all SQL PA DB2 objects are dropped, specify YES. To drop all SQL PA DB2 objects except the database, specify NO.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WLM environment for sample stored procs</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>DSN8WLMR</td>
<td></td>
</tr>
<tr>
<td>Specify the name of the WLM environment to use for the sample stored procedures.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(DSGROUP) Number of members in data sharing group</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>The DSGROUP system parameter indicates the number of members in the data sharing group. This parameter is used with the DATASHR parameter to adjust the overall processing power of the system to reflect excess processing time that is incurred by data sharing operations. The result of using these two parameters is a more accurate cost assessment.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(DATASHR) Workload percentage using data sharing</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>The DATASHR system parameter specifies the average percentage of the DB2 workload that participates in data sharing. This parameter is used with the DSGROUP parameter to adjust the overall processing power of the system to reflect the excess processing time that is incurred by data sharing operations. The result of using these parameters is a more accurate cost assessment.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SQL PA DB2 General Information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Discovered</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(BUFFHIT) Buffer hit percentage</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>The BUFFHIT user parameter indicates the ratio of physical to logical I/O, expressed as a percentage from 0 to 100. Specify a value for the percentage of pages that are found in the buffer pool (that is, for which disk I/O is not necessary). For example, specifying 20 means that 20% of the time the required pages are found in the buffer pool, whereas 80% of the time a physical I/O to disk is required. Specifying 000 means that 100% of time the I/O goes to disk, and no pages are assumed to be in the pool.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(BUFFERS) The number of 4 KB buffers in 4 KB buffer pool</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>Set the BUFFERS system parameter to the total number of 4 KB buffers in the 4 KB buffer pool of the dominant application. This value is used to set the number of pages per block for sequential prefetch and other I/O calculations. Specify the most prominent 4 KB buffer pool here, even if several are used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(BUFF08K) The number of 8 KB buffers in 8 KB buffer pool</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>Set the BUFF08K system parameter to the total number of 8 KB buffers in the 8 KB buffer pool of the dominant application. This value is used to set the number of pages per block for sequential prefetch and other I/O calculations. Specify the most prominent 8 KB buffer pool here, even if several are used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(BUFF16K) The number of 16 KB buffers in 16 KB buffer pool</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>250</td>
<td></td>
</tr>
</tbody>
</table>
### SQL PA DB2 General Information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(BUFF32K) The number of 32 KB buffers in 32 KB buffer pool</td>
<td>Yes</td>
<td>Yes</td>
<td>100</td>
<td>blank</td>
</tr>
<tr>
<td>Set the BUFF32K system parameter to the total number of 32 KB buffers in the 32 KB buffer pool of the dominant application. This value is used to set the number of pages per block for sequential prefetch and other I/O calculations. Specify the most prominent 32 KB buffer pool here, even if several are used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SQL PA Installation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customization ID to use for setup</td>
<td>Yes</td>
<td>Yes</td>
<td>ANLUSER0</td>
<td>blank</td>
</tr>
<tr>
<td>The DB2 user ID that is to used to customize SQL PA. This ID owns the SQL PA objects that are created for the customization and must have SYSADM authority.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customized library HLQ</td>
<td>Yes</td>
<td>Yes</td>
<td>SQLPA.ANL420</td>
<td>blank</td>
</tr>
<tr>
<td>The high-level qualifier of the SQL PA customized libraries of SANLDA, SANLJCL, and SANLPARM. If the parameter is not specified, it defaults to the high-level qualifier of the SQL PA installation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customized library volume</td>
<td>Yes</td>
<td></td>
<td>blank</td>
<td>blank</td>
</tr>
<tr>
<td>The name of the volume of SANLPARM, SANLJCL, SANLDA, SANLLODS and SANLVCLS customized libraries. The default is blank, which means that SMS will select the volume.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load library lower level qualifier</td>
<td>Yes</td>
<td>Yes</td>
<td>SANLLOAD</td>
<td>blank</td>
</tr>
<tr>
<td>This parameter indicates the lower level of qualification, if the load modules are copied to a library in the target libraries other than the default library of SANLLOAD. The default is SANLLOAD.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secondary authids for SQL PA</td>
<td>Yes</td>
<td>Yes</td>
<td>blank</td>
<td>blank</td>
</tr>
<tr>
<td>Specifies the optional secondary authorization IDs to use as the owners of the generic plan tables.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Required?</td>
<td>Discovered?</td>
<td>Default value</td>
<td>Your value</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>QMF user ID</td>
<td>Yes</td>
<td>Yes</td>
<td>blank</td>
<td>blank</td>
</tr>
<tr>
<td>(USRPARM) Allow users to have a copy of the user parameters</td>
<td>Yes</td>
<td>No</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>(PROCESS) Enable automatic version recognition</td>
<td>Yes</td>
<td>Yes</td>
<td>+OFF+</td>
<td>+OFF+</td>
</tr>
<tr>
<td>(CONNECT) Attach facility</td>
<td>Yes</td>
<td>Yes</td>
<td>CAF</td>
<td>CAF</td>
</tr>
</tbody>
</table>

If you have QMF installed, indicate the owner of the QMF DB2 tables. If you do not have QMF installed, leave the parameter value blank.

The USRPARM system parameter authorizes the use of the user-level user parameters.

Set the PROCESS user parameter to ASIS to turn off the automatic DB2 version recognition capability. If you have coded a VERSION parameter that specifies a DB2 release that is different than the actual version under which the program is running, the VERSION parameter is reset automatically unless the ASIS value is set. When the PROCESS parameter is set to +OFF+, ANL2021W is displayed when the VERSION parameter is reset.

DB2 SQL Performance Analyzer lists the total MIPS rate for a processor, and the calculated speed per engine, which is the number that is used for most CPU computations. DB2 SQL Performance Analyzer uses the (CONNECT) Attach facility parameter to determine which attachment facility has been chosen to connect to DB2. The Detail Trace Report includes information about what DB2 SQL Performance Analyzer assumes is the processor usage for that connection, which is included once for each query issued.
### SQL PA Installation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STORAGE</strong> Storage device</td>
<td>Yes</td>
<td>Yes</td>
<td>3390-3</td>
<td></td>
</tr>
</tbody>
</table>

The STORAGE user parameter reflects the most prominent DASD storage medium for an application's databases. If your system uses several device types, select the device that holds most of your databases. The default value is 3390-3.

Valid values for disk device types are 3390, 3390-1, 3390-2, 3390-3, 3390-9, 6390, 6390-3, 6390-5, 6390-9, 7390-1, 7390-2, 7390-3, or 7390-F.

Valid values for solid-state devices are 4305, 4380, 6110, 6110-1, 6110-2, 6110-4, 6680-1, 7900, ICEBRG, STK, EMCX, EMCBAR, EMCELT, RAMAC, RAMAC2, RAMAC3, ESSE20, ESSF20, ESS800, and ESS80T (turbo).

If several device types are on your system, specify the device that holds most of your databases.

| **NEWSTOR** Newly defined storage device | Yes | Yes | blank |  |

The NEWSTOR user parameter indicates that you are defining a new storage device. Although a number of device types are already defined, including ESS, RAMAC, EMC, ICEBERG, and other solid state devices, you can define your own device type by using this parameter. Follow this parameter with the NEWSEEK, NEWROTO, and NEWXFER parameters to define the new device. Either this parameter or a STORAGE NEWDSK setting indicates that new disk storage is being defined.

| **NEWROTA** Newly defined rotational delay | Yes | blank |  |
**SQL PA Installation Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NEWSEEK) Newly defined seek time</td>
<td>Yes</td>
<td>Yes</td>
<td>blank</td>
<td></td>
</tr>
<tr>
<td>The NEWSEEK user parameter specifies the average seek time in seconds for the definition of a new storage device. This parameter is effective only when the NEWSTOR parameter is also set or when the STORAGE user parameter is set to NEWDSK.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(NEWXFER) Newly defined transfer rate</td>
<td>Yes</td>
<td>Yes</td>
<td>blank</td>
<td></td>
</tr>
<tr>
<td>The NEWXFER user parameter specifies the average transfer rate of the device over the channel in kilobytes per second. This parameter is effective only when the NEWSTOR parameter is also set or when the STORAGE parameter is set to NEWDSK.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ENGINES) Number of engines</td>
<td>Yes</td>
<td>Yes</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>The ENGINES system parameter indicates the total number of engines in this processor complex.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(LPARENG) Number of engines dedicated to DB2</td>
<td>Yes</td>
<td>Yes</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>The LPARENG system parameter specifies the number of engines from the processor that are dedicated to the logical partition on which DB2 is running. You can specify the value in terms of partial processors, for example 1.85. This parameter addresses the situation in which a single physical processor complex is divided into multiple system images. The setting of this parameter should reflect the logical system image on which the DB2 target host system is running. This parameter is used to isolate the maximum single engine speed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ESACOMP) Hardware data compression in use</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>The ESACOMP system parameter specifies whether this machine has a hardware data compression feature. For table spaces that are data compression candidates, the choice of hardware or software significantly affects the path length of encoding and decoding rows.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SQL PA Installation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ESASORT) Hardware assisted sort in use</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

The ESASORT system parameter specifies whether the DB2 hardware assisted sort facility, or its equivalent, is available on this processor. The DB2 sort algorithms for SQL PA are influenced by this parameter.

### SQL PA Batch Processing Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>First line of the jobcard of batch JCL</td>
<td>Yes</td>
<td>Yes</td>
<td>//ANLBATCH JOB ,</td>
<td></td>
</tr>
<tr>
<td>Second line of the jobcard of batch JCL</td>
<td>Yes</td>
<td>Yes</td>
<td>//USER=&amp;SYSUID,NOTIFY=&amp;SYSUID</td>
<td></td>
</tr>
<tr>
<td>Third line of the jobcard of batch JCL</td>
<td>Yes</td>
<td>Yes</td>
<td>//MSGLEVEL=(1,1),REGION=0M,T</td>
<td></td>
</tr>
<tr>
<td>Fourth line of the jobcard of batch JCL</td>
<td>Yes</td>
<td>Yes</td>
<td>/<em>JOBPARM S=</em></td>
<td></td>
</tr>
<tr>
<td>(QUALIFY) Default qualifier for objects</td>
<td>Yes</td>
<td>Yes</td>
<td>blank</td>
<td></td>
</tr>
</tbody>
</table>

The QUALIFY user parameter specifies the qualifier that will be used to qualify unqualified objects in an SQL statement. If no value is specified, the current SQL ID will be used.

### SQL PA Cost Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(CPUCOST) Cost of one hour of CPU time</td>
<td>Yes</td>
<td>Yes</td>
<td>500</td>
<td></td>
</tr>
</tbody>
</table>

Set the CPUCOST system parameter to the cost of one hour of CPU time, in the national currency, for this configuration. This parameter is used by SQL PA for charge back and financial cost determination.
### SQL PA Cost Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(IOSCOST) Cost of 1000 I/O calls</td>
<td>Yes</td>
<td>Yes</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Set the IOSCOST system parameter to the cost of 1000 I/O calls, in the national currency, for this configuration. This parameter is used by SQL PA for charge back and financial cost determination.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(TIMCOST) Cost of one hour of connect time</td>
<td>Yes</td>
<td>Yes</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Set the TIMCOST system parameter to the cost of one hour of connect time, in the national currency, for this configuration. This parameter is used by SQL PA for charge back and financial cost determination.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(MONEYIS) National currency</td>
<td>Yes</td>
<td>Yes</td>
<td>DOLLARS</td>
<td></td>
</tr>
<tr>
<td>Set the MONEYIS system parameter to the national currency, in descriptive form, such as DOLLARS, POUNDS, STERLING, DRACHMA, or KRONA.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(CURRSYM) Currency symbol</td>
<td>Yes</td>
<td>Yes</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>Set the CURRSYM system parameter to the single character symbol that represents the national currency that corresponds to the MONEYIS parameter.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SQL PA EXPLAIN Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SETPLAN) Allow users to set the plan table owner</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>The SETPLAN system parameter authorizes the use of the USEPLAN user parameter to indicate the qualifier of the plan table if its value is set to YES. If this parameter is set to NO, the value of USEPLAN is ignored, and the owner of the plan table defaults to the normal qualification.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Required?</td>
<td>Discovered?</td>
<td>Default value</td>
<td>Your value</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>(KEEPLAN) Retain plan table information</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td><strong>The KEEPLAN user parameter</strong> indicates whether plan records that are owned by the value of the USEPLAN parameter are retained when SQL PA runs in batch. SQL PA deletes all plan records at the conclusion of each run, and it deletes all plan records with QUERYNO greater than 100 M on startup. If this parameter is set to YES, the deletion of plan records in your permanent plan tables is prevented. This parameter does not affect online operation, and it does not affect the generic plan tables.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(PRECISE) User DB2 optimizer estimates</td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td><strong>The PRECISE user parameter provides an extra level of precision to the cost estimate. If this parameter is set to YES, SQL PA replaces its internal cost estimate and QUNIT result with the values from DB2. It also uses the optimizer's path length in calculations. If this parameter is set to ALL, the cost estimate includes additional factors and the Predicate Analysis report is generated.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(NLSCODE) DBCS or mixed data names in SQL</td>
<td>Yes</td>
<td>Yes</td>
<td>+OFF+</td>
<td></td>
</tr>
<tr>
<td><strong>The NLSCODE user parameter enables the processing of SQL statements that include DBCS or mixed data names, such as those that are found in the Korean, Japanese, and Chinese languages. If this parameter is set to +OFF+, SQL is converted to uppercase before processing. This translation might alter the DBCS and mixed data names. The lowercase-to-uppercase translation is not performed with any other setting.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Required?</td>
<td>Discovered?</td>
<td>Default value</td>
<td>Your value</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>(PROCESS) DBRM sequence numbers out of order</td>
<td>Yes</td>
<td>Yes</td>
<td>+OFF+</td>
<td></td>
</tr>
<tr>
<td>Set the PROCESS user parameter to NOSEQ if your DBRM was created or modified by third-party software products that have caused the sequence numbers of the statements to be out of order. Without this setting, message ANL2003E will be issued because of synchronization errors that result from the numbers being out of sequence.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(DEGREES) Use parallel processing</td>
<td>Yes</td>
<td>Yes</td>
<td>ONE</td>
<td></td>
</tr>
<tr>
<td>Set the DEGREES user parameter to specify whether parallel processing will be considered during the statement evaluation when the batch interface is used. Set DEGREES to ANY when parallel processing is enabled. Set DEGREES to ONE when parallel processing is not enabled.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(DSN8EXP) Use DB2-supplied stored procedure</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Set the DSN8EXP user parameter to specify whether EXPLAINs are run by using the DB2-supplied stored procedure (DSNAEXP) or by embedded SQL. If the value is set to YES, DSNAEXP is used if it is installed and available; otherwise, DSN8EXP is used.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(DELIMIT) COBOL precompiler option QUOTESQL used</td>
<td>Yes</td>
<td>Yes</td>
<td>+OFF+</td>
<td></td>
</tr>
<tr>
<td>Set the DELIMIT user parameter to indicate that the COBOL precompiler option of QUOTESQL was used. If QUOTESQL was used and DELIMIT is not set to QUOTE, a -206 SQLCODE is issued.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(REFRESH) Consider materialized query tables</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>The REFRESH user parameter allows the optimizer to consider materialized query tables (MQTs) as candidates for access path selection, if the value is set to ALL or ANY. If this parameter is set to NO, MQTs are not considered.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SQL PA EXPLAIN Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(DYNAMIC) Dynamic statement caching</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set the DYNAMIC system parameter to specify whether dynamic statement caching is used. Caching dynamic SQL statements can reduce the excess processing time that is associated with preparing dynamic SQL for subsequent operations.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SQL PA Report Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ACCTYPE) Plan report access type filter</td>
<td>Yes</td>
<td>Yes</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set the ACCTYPE user parameter to filter the statements of plans and packages for the Easy Explain Explanation report according to the types of access path. Specify ALL or A to show all statements. Use HASH or H to show only statements that use a hash access path. Specify MATCHING or M to show only statements that use a matching index scan access path. Specify NONMATCH or N to show only statements that use a non-matching index scan access path. Specify TABSCAN or T to show only statements that use a table space scan access path.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(FORMAT) Format SQL in plan report</td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The FORMAT user parameter indicates how Easy Explain formats the SQL statements of the plans and packages. Set this parameter to YES if you want each SQL keyword, such as SELECT, INTO, FROM, and WHERE, to start on a new line. Set this parameter to NO if you want only certain keywords, such as SELECT and UNION, to start on a new line. Setting this parameter to NO consumes less page space.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SQL PA Report Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(HOSTVAR) Host variables in plan report</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>The HOSTVAR user parameter indicates whether Easy Explain lists the host variable information of the plans and packages. If this parameter is set to YES, the Easy Explain report lists up to 100 host variables and their definitions. If this parameter is set to NO, no host variable definitions are listed in the Easy Explain report.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(INDEX) Plan report index level</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>The INDEX user parameter controls the level of index information in the Easy Explain report. When this parameter is set to ALL or CONDENSE, information for all indexes that were created for the table is shown. When this parameter is set to NO, no index information is shown. When this parameter is set to TABLESPACE or YES, detailed index information is shown.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(PLANINF) Plan report plan info level</td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>The PLANINF user parameter controls the printing of the plan and DBRM/package report after the first miniplan in a plan of the Easy Explain report. If this parameter is set to YES, the plan and DBRM/package report are printed if specified in the authorization table. If this parameter is set to MINI, a short report that shows the plan parameters is printed. If this parameter is set to NO, a plan or DBRM/package report is not printed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SQL PA Report Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(LEVEL) Plan report format</td>
<td>Yes</td>
<td>Yes</td>
<td>DETAIL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The LEVEL user parameter controls the level of information in the plan table report. DETAIL shows a detailed report. SQLSUMMARY shows only the SQL statement and access path messages. SUMMARY shows only the summary page with one line for each SQL statement. TABULAR shows a brief report with plan table information in tabular format. For more information, see the following options: ACCESSPATH, ALLINDEX, KEY, NOCATALOG, NOKEYS, TINO, TIYES, XOFF, and XON.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(TURNOFF) Exclude messages from output reports</td>
<td>Yes</td>
<td>Yes</td>
<td>blank</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The TURNOFF user parameter excludes specified messages from the SQL PA output. Specify the 4-digit numerical message number to turn off the message. You can enter as many TURNOFF parameters as you want, but you must enter each one on a separate line. The default is to show all messages.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(REPORTS) Level of SQL PA reports</td>
<td>Yes</td>
<td>Yes</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The REPORTS user parameter specifies the level of reporting. If this parameter is set to LOG, the cost report will be generated, which provides a summary of query cost plus any warning messages. If this parameter is set to EXP, the Enhanced Explain report is generated with the cost report. If this parameter is set to DET or ALL, the detailed trace information is generated with the cost report and the Enhanced Explain report.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SQL PA Report Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Discovered</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SHOWALT</strong> Show alternate indexes</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>QLIMSQL</strong> Display SQL error in query limits report</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MESSAGE</strong> Include message text in reports</td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OBJECTS</strong> Display tables and indexes in SQL report</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DSPPARM</strong> Display parameters in output reports</td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SHOWALT**
Set the SHOWALT user parameter to YES to display information about alternate indexes that are not selected by the optimizer in the Enhanced Explain and detail trace reports. The information is presented for access paths of I, N, and R only, and is not presented for multiple index or direct access using row ID. The information includes pertinent catalog statistics about the size and shape of each alternate index, and information about the index key's columns.

**QLIMSQL**
Set the QLIMSQL user parameter to YES to display an SQL statement that received a negative SQLCODE in the QLIMITS report.

**MESSAGE**
The MESSAGE user parameter determines whether the message text will be included in the SQL PA output. If this parameter is set to NO, only the message ID will be displayed.

**OBJECTS**
The OBJECTS user parameter determines whether the list of objects (tables and indexes) that are used will be included in the SQL report. If this parameter is set to YES, the list of objects is displayed. If this parameter is set to NO, the list of objects is not displayed.

**DSPPARM**
The DSPPARM user parameter determines whether the SQL PA parameters and their values are displayed at the top of the enhanced explain and detail trace reports. If this parameter is set to YES, the parameters are displayed. If this parameter is set to NO, the parameters do not display.
### SQL PA Report Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(DSPVARS) Display host variables in SQL</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>The DSPVARS user parameter determines whether the host variable names or parameter markers are displayed in SQL statements in the cost reports. If this parameter is set to YES, the host variables are displayed. If this parameter is set to NO, the parameter markers are displayed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(NUMBERS) Input file created with NUMBERS ON</td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>The NUMBERS user parameter specifies whether the input file was created with NUMBERS ON.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SQL PA Message Handling Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ADVISOR) ADVISOR level for output reports</td>
<td>Yes</td>
<td>No</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>The ADVISOR user parameter displays additional information about the performance and design of each query and about the tables and indexes that are accessed by the query. Warnings and Alerts are always displayed, even when ADVISOR is set to NO. Notes and Recommendations are displayed when ADVISOR is set to YES, which can reveal potential performance tuning opportunities. A value of ALL is ideal for those who want to receive Guideline messages and confirmation that tasks have concluded successfully.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(RETCODE) Return code reflects SQL PA warnings</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>The RETCODE user parameter governs whether the return code reflects the warnings, error messages, or cost overruns from SQL PA. If this parameter is set to NO, the program return code reflects the normal processing return code of the SQL PA programs. If this parameter is set to YES, values of 4, 8, 12, or 16 are returned as program return codes from normal operation that encountered SQL PA messages.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SQL PA Message Handling Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ALLPART) Warn when TS scan reads all partitions</td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Set the ALLPART user parameter to YES to be notified when a partitioned table space scan reads all partitions in the table (the limited range scan is not used). Typically, reading all partitions in a table is costly.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(NONONIX) Warn on non-matching index scan</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Set the NONONIX user parameter to YES to be notified of every non-matching index scan operation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(NOSTATS) Warn when statistics not set</td>
<td>Yes</td>
<td>Yes</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Set the NOSTATS user parameter to YES to be notified when the default was used for the statistics of any table, index, table space, or column because the catalog statistics were not set.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(NOTSCAN) Warn on table space scan</td>
<td>Yes</td>
<td>Yes</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Set the NOTSCAN user parameter to YES to be notified of every table space scan operation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SQL PA Warning Message Enablement Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(CPUTIME) Maximum CPU time</td>
<td>Yes</td>
<td>Yes</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>The CPUTIME system parameter indicates the maximum CPU time, in seconds, that is permitted before the SQL statement is flagged as exceeding the installation's CPU time limit. The maximum value is 86400, which is equivalent to 24 hours. A setting of 0 results in unlimited CPU time.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(COSTING) Maximum query cost</td>
<td>Yes</td>
<td>Yes</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>The COSTING system parameter indicates the maximum monetary value, in your currency, that is permitted before the SQL statement is flagged as exceeding the installation's monetary limit. The maximum value is 999999999. A setting of 0 results in no monetary limit.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Required?</td>
<td>Discovered?</td>
<td>Default value</td>
<td>Your value</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>(IOCALLS) Maximum I/O calls</td>
<td>Yes</td>
<td>Yes</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>The IOCALLS system parameter indicates the maximum physical I/O calls that are permitted before the SQL statement is flagged as exceeding the installation’s physical I/O limit. The maximum value is 999999999. A setting of 0 results in unlimited physical I/O.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(COSTQUN) Maximum query units</td>
<td>Yes</td>
<td>Yes</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>The COSTQUN system parameter indicates the maximum number of QUNITs that are permitted before the SQL statement is flagged as exceeding the installation’s QUNIT limit. The maximum value is 999999999. A setting of 0 results in no QUNIT limit.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ELAPSED) Maximum elapsed time</td>
<td>Yes</td>
<td>Yes</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>The ELAPSED system parameter indicates the maximum elapsed time, in seconds, that is permitted before the SQL statement is flagged as exceeding the installation’s elapsed time limit. The maximum value is 86400, which is equivalent to 24 hours. A setting of 0 results in unlimited elapsed time.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(INLISTS) Maximum number of IN (LIST) elements</td>
<td>Yes</td>
<td>Yes</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>The INLISTS user parameter specifies the maximum number of elements that are shown in an IN (LIST) predicate. If the maximum number of elements is exceeded, SQL PA issues a notification message.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ISCANPG) Maximum IX leaf pages read (matching scan)</td>
<td>Yes</td>
<td></td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>The ISCANPG user parameter indicates the maximum number of index leaf pages that are read with a matching index scan. If the maximum number of pages is exceeded, SQL PA issues a notification message.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SQL PA Warning Message Enablement Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Discovered?</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NONIXPG) Max IX leaf pages read (non-match scan)</td>
<td>Yes</td>
<td>Yes</td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td>The NONIXPG user parameter indicates the maximum number of index leaf pages that are read with a non-matching index scan. If the maximum number of pages is exceeded, SQL PA issues a notification message.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(IXUPDAT) Maximum indexes on table update</td>
<td>Yes</td>
<td>Yes</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>The IXUPDAT user parameter indicates the maximum number of indexes that can exist on a table that is being updated. If the maximum number of indexes is exceeded, SQL PA issues a notification message.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(JOINTAB) Maximum tables to be joined</td>
<td>Yes</td>
<td>Yes</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>The JOINTAB user parameter indicates the maximum number of tables that can be joined together. If the maximum number of tables is exceeded, SQL PA issues a notification message.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(MATCOLS) Minimum index columns matched</td>
<td>Yes</td>
<td>Yes</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>The MATCOLS user parameter limits the number of columns in an index that are used in a matching index scan. The parameter value expresses the fraction of columns in the index key. For example, the default value of 0.50 states that at least half of the columns in the index key should be used in a matching index scan. If not, SQL PA will issue a notification message that indicates that the index is not being used to its fullest potential.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(PREDPCT) Maximum predicate filtering percentage</td>
<td>Yes</td>
<td>Yes</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>The PREDPCT user parameter limits the fraction of the number of total rows to be retained after predicate filtering. When this limit is exceeded, a notification message is issued that indicates that the predicates are not effectively restrictive. For example the default value of 0.15 states that if more than 15% of the rows pass filtering, SQL PA will issue a message.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Required?</td>
<td>Discovered?</td>
<td>Default value</td>
<td>Your value</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>(TSCANPG) Maximum data pages for table space scan</td>
<td>Yes</td>
<td>Yes</td>
<td>50000</td>
<td></td>
</tr>
</tbody>
</table>

The TSCANPG user parameter indicates the limit of the number of data pages that are read with a table space scan. If the maximum number of pages is exceeded, SQL PA issues a notification message.

---

## Starting and preparing Tools Customizer for use

Use the provided REXX EXEC to start Tools Customizer. The first time that you use Tools Customizer, you must modify the settings that Tools Customizer uses to customize DB2 SQL Performance Analyzer, DB2 Autonomics Director, IZ Sample Tool.

### Starting Tools Customizer

Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel.

### Before you begin

Tools Customizer must be SMP/E installed. You must know the high-level qualifier of where the Tools Customizer libraries reside. The high-level qualifier is considered to be all the segments of the data set name except the lowest-level qualifier, which is SCCQEXEC.

**Attention:** Ensure that Tools Customizer load libraries are not APF authorized. APF authorizing Tools Customizer libraries results in an abend.

### About this task

To run the REXX EXEC, you must either change the placeholder in the EXEC for the high-level qualifier of the Tools Customizer EXEC library or pass the high-level qualifier as a parameter when you run the EXEC. The REXX EXEC is in the CCQTCZ member of the EXEC library.

### Procedure

1. Optional: Change the placeholder for the high-level qualifier in the REXX EXEC:
   a. Find the EXEC library data set for Tools Customizer. The name of the data set is `high_level_qualifier.SCCQEXEC`.
   b. Edit data set member CCQTCZ and replace the `<TCZ HLQ>` string with the high-level qualifier of the EXEC library data set. For example, if the name of the Tools Customizer EXEC library is `CCQTCZ.USABSAND.SCCQEXEC`, replace `<TCZ HLQ>` with `CCQTCZ.USABSAND`.

2. Run the REXX EXEC (CCQTCZ):
a. From the ISPF Primary Option Menu, select option 6. The ISPF Command Shell panel is displayed.
b. Specify the EX command to run the REXX EXEC. For example, if the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC and you changed the placeholder for the high-level qualifier in the REXX EXEC, specify: EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)'
   If you did not change the placeholder for the high-level qualifier in the REXX EXEC, specify: EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)'
   'CCQTCZ.USABSAND'

Results

The IBM Customizer Tools for z/OS main menu panel is displayed.

What to do next

If you are running Tools Customizer for the first time, you must modify the Tools Customizer user settings. If you have already set the Tools Customizer user settings, either customize or recustomize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

Modifying Tools Customizer user settings

Before you can customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool with Tools Customizer, you must review the settings that Tools Customizer uses. You might have to change the default values to suit your environment. In most cases, you can change the Tools Customizer values at any time. For example, after you have customized DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool and are customizing a different product or solution pack, you might have to change the settings.

Procedure

1. On the IBM Tools Customizer for z/OS main panel (CCQPHME), specify option 0, User settings for Tools Customizer. The Tools Customizer Settings panel (CCQPSET) is displayed, as shown in the following figure:

   ![The Tools Customizer Settings panel (CCQPSET)](image)

   Figure 4. The Tools Customizer Settings panel (CCQPSET)
2. Review the values for the following required fields. Use the default value or specify your own value. You must have appropriate read and write access to the data sets that are specified.

**Customization library qualifier**

The high-level qualifier that is used as the prefix for the customization library. The customization library is a data set in which the generated jobs to customize DB2 SQL Performance Analyzer, DB2 Autonomics Director, and Mz Sample Tool are stored. Write access to this qualifier is required.

For each product to be customized, the first value that is specified for the qualifier is always used, even if you change it after you have generated the customization jobs. For example, if you customize a product and then specify a new qualifier for recustomization, although the new qualifier is saved and displayed, the original value is used.

To maintain multiple instances of Tools Customizer, specify a unique customization library qualifier for each instance of Tools Customizer.

Data set names that exceed 42 characters must be enclosed in single quotation marks (').

**Use DB2 group attach**

Determines the value that is used in the CONNECT statements in the generated customization jobs. Specify YES for data sharing environments, which causes the group attach name to be used. Specifying NO, in most cases, causes the SSID to be used in the DB2 CONNECT statement.

**Important:** This field has no effect when you are customizing a product on a DB2 subsystem that is not a member of a data sharing group. In this case, the DB2 subsystem ID (SSID) is always used in the CONNECT statements in the generated customization jobs.

When you are customizing a product on a DB2 subsystem that is a member of a data sharing group, how the DB2 subsystem is defined and the value of the Use DB2 group attach field determines the value that is used in the CONNECT statements in the generated jobs. The following table shows whether the SSID or the group attach name is used:

<table>
<thead>
<tr>
<th>DB2 subsystem definition</th>
<th>Value of the Use DB2 group attach field</th>
<th>Value that is used in the CONNECT statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The DB2 subsystem is defined with an SSID.</td>
<td>Yes</td>
<td>Group attach name</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>SSID¹</td>
</tr>
<tr>
<td>The DB2 subsystem is not defined with an SSID.</td>
<td>Yes or No</td>
<td>Group attach name</td>
</tr>
</tbody>
</table>

**Note 1:** If you generate jobs for multiple DB2 subsystems that are defined with an SSID and belong to the same data sharing group, the SSID of the first DB2 subsystem that is selected is used.

For example, assume that on the Customizer Workplace panel, you generated jobs for the following DB2 subsystems:

- V91C, which is a stand-alone DB2 subsystem
- V91A, which is a DB2 subsystem that is a member of data sharing group DSG1
- A DB2 subsystem that was not defined with an SSID that is a member of data sharing group DSGA

The following figure shows how these DB2 entries might be listed on the Customizer Workplace panel:

The following table shows which values are used in the CONNECT statements in the generated jobs, based on the value of the Use DB2 group attach field.

<table>
<thead>
<tr>
<th>SSID</th>
<th>GrpAttach</th>
<th>Value of the Use DB2 group attach field</th>
<th>Value that is used in the CONNECT statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>V91C</td>
<td>--</td>
<td>Yes</td>
<td>SSID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td>SSID</td>
</tr>
<tr>
<td>V91A</td>
<td>DSG1</td>
<td>Yes</td>
<td>Group attach name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td>SSID</td>
</tr>
<tr>
<td>--</td>
<td>DSGA</td>
<td>Yes</td>
<td>Group attach name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td>Group attach name</td>
</tr>
</tbody>
</table>

Tools Customizer metadata library

The name of the data set that contains the metadata that is used to display the DB2 and LPAR parameters. The parameters that are displayed on the LPAR Parameters panel and the DB2 Parameters panel depend on the parameters that you define and the tasks and steps that you select on the Product Parameters panel for the product that you are customizing. For example, the DB2 parameters that are required, based on the selected tasks and steps, are displayed on the DB2 Parameters panel, and you can edit them. If they are not required, they are not displayed. Read access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Discover output data set

The name of the data set in which the output from the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC is stored. Each product has its own Discover EXEC. The Discover EXEC retrieves the product, LPAR, and DB2 parameters from a previously customized product. Write access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Data store data set

The name of the data set where Tools Customizer stores information about product, LPAR, and DB2 parameter values. Information about which products are associated with which DB2 entries (DB2
subsystems, DB2 group attach names, and DB2 data sharing members) is also stored in this data set. Data set names that exceed 42 characters must be enclosed in single quotation marks ('). The specified data store data set can be used with only one invocation of Tools Customizer at a time. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

**User job card settings for customization jobs**

The job card information to be inserted into the generated jobs for customizing a product. The default value is the job statement information from the ISPF Batch Selection panel.

The first line of the job card automatically begins with the following information:

```
// JOB
```

where characters 3 - 10 are reserved by Tools Customizer for the job name and includes a blank space after JOB. This name cannot be edited. Information that you specify on the first line of the job card cannot exceed 57 characters. This character limit includes a continuation character. All other lines of the job card cannot exceed 72 characters.

3. Press End to save and exit. If the Discover output data set and the data store data set that you specified do not exist, Tools Customizer creates them.

**Important:** If the ISPF sessions unexpectedly ends before you exit Tools Customizer, the fields on the Tools Customizer Settings panel (CCQPSET) will be repopulated with default values, and you will be required to review them or specify new values again.

**Results**

The values are saved, and the IBM Tools Customizer for z/OS main menu panel (CCQPHME) is displayed again.

**What to do next**

You are ready to customize or recustomize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool or to change parameter settings.

**Related concepts:**

- “Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool” on page 65
- Using Tools Customizer to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool consists of identifying the product to customize; defining any required DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, LPAR, and DB2 parameters; generating the customization jobs; and submitting the jobs.

**Hiding and displaying panel text**

After you are familiar with Tools Customizer, you might want to hide the instructions and some of the basic descriptions that are displayed by default on Tools Customizer panels.
About this task

By using the OPTIONS command, you can choose to show or hide the following information on Tools Customizer panels:

- The instructions on all panels
- The Product to Customize section on the Customizer Workplace panel (CCQPWRK)
- The Usage Notes section on the Product Parameters panel (CCQPRRD), the LPAR Parameters panel (CCQPLPR), and the DB2 Parameters panel (CCQPDB2).

By hiding this information, more data can be displayed on the panels. Later, you can redisplay this information also by using the OPTIONS command.

Procedure

1. On any Tools Customizer panel, issue the OPTIONS command. The Panel Display Options panel (CCQPOPT) is displayed, as shown in the following figure. By default, all options are preselected with a slash (/) to be shown.

![Panel Display Options panel (CCQPOPT)](image)

2. To hide any of the options, remove the slash, and press Enter.

Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool

Using Tools Customizer to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool consists of identifying the product to customize; defining any required DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, LPAR, and DB2 parameters; generating the customization jobs; and submitting the jobs.

Customization roadmaps describe the steps that you must complete to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool. Separate roadmaps are provided for the three most common types of customizations.

Use the following table to determine which roadmap corresponds to your environment.

<table>
<thead>
<tr>
<th>Environment description</th>
<th>Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>You do not have a customized version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, and you need to customize it for the first time.</td>
<td>&quot;Roadmap: Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool for the first time&quot; on page 66</td>
</tr>
</tbody>
</table>
Table 4. Customization roadmaps (continued)

<table>
<thead>
<tr>
<th>Environment description</th>
<th>Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>You have already customized a version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, and you want to use the same parameter values to customize a different version.</td>
<td>“Roadmap: Customizing a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool from a previous customization” on page 67</td>
</tr>
<tr>
<td>You have a customized version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, but you want to change one or more parameter values.</td>
<td>“Roadmap: Recustomizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool” on page 68</td>
</tr>
</tbody>
</table>

Roadmap: Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool for the first time

This roadmap lists and describes the steps that are required to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool for the first time.

If you are customizing a previous version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, see “Roadmap: Customizing a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool from a previous customization” on page 67.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- The LPAR ISPF libraries that are required to submit the jobs are known.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool for the first time.

Table 5. Steps for customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool for the first time

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specify the product metadata library for the product that you want to customize. The name of this library is hlq.SANLDYXIMZDENU.</td>
<td>“Specifying the metadata library for the product to customize” on page 70</td>
</tr>
<tr>
<td>2</td>
<td>Create new DB2 entries and associate them with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.</td>
<td>“Creating and associating DB2 entries” on page 74</td>
</tr>
<tr>
<td>3</td>
<td>Define the required parameters.</td>
<td>“Defining parameters” on page 77</td>
</tr>
<tr>
<td>4</td>
<td>Generate the customization jobs for the product or for the DB2 entries on which DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool is ready to be customized.</td>
<td>“Generating customization jobs” on page 83</td>
</tr>
<tr>
<td>5</td>
<td>Submit the generated customization jobs.</td>
<td>“Submitting customization jobs” on page 84</td>
</tr>
</tbody>
</table>
The following table lists some of the common administrative tasks that you might need to do during the customization process.

### Table 6. Administrative tasks

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse the different types of parameters.</td>
<td>“Browsing parameters” on page 87</td>
</tr>
<tr>
<td>Copy an existing DB2 entry to the list of DB2 entries on which DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool can be customized.</td>
<td>“Copying DB2 entries” on page 87</td>
</tr>
<tr>
<td>Remove one or more DB2 entries from the associated list.</td>
<td>“Removing DB2 entries” on page 88</td>
</tr>
<tr>
<td>Delete one or more DB2 entries from the master list.</td>
<td>“Deleting DB2 entries” on page 89</td>
</tr>
<tr>
<td>Display a list of customization jobs that have been previously generated.</td>
<td>“Displaying customization jobs” on page 89</td>
</tr>
<tr>
<td>Maintain the customization jobs in the customization library.</td>
<td>“Maintaining customization jobs” on page 90</td>
</tr>
</tbody>
</table>

### Roadmap: Customizing a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool from a previous customization

This roadmap lists and describes the steps for customizing a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool based on the existing customization values of a previous version of the same product.

Use this roadmap even if the previous version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool was not customized by using Tools Customizer.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool from a previous customization.

### Table 7. Steps for customizing a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool from a previous customization

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specify the product metadata library for the product that you want to customize. The name of this library is hlq.SANLDYXIMZDENU.</td>
<td>“Specifying the metadata library for the product to customize” on page 70</td>
</tr>
</tbody>
</table>
Table 7. Steps for customizing a new version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool from a previous customization (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Use the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool Discover EXEC to discover information about the version of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool that you previously customized manually.</td>
<td>“Discovering DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool information automatically” on page 72</td>
</tr>
<tr>
<td>3</td>
<td>Define the required parameters.</td>
<td>“Defining parameters” on page 77</td>
</tr>
<tr>
<td>4</td>
<td>Generate the customization jobs for the product or for the DB2 entries on which DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool is ready to be customized.</td>
<td>“Generating customization jobs” on page 83</td>
</tr>
<tr>
<td>5</td>
<td>Submit the generated customization jobs.</td>
<td>“Submitting customization jobs” on page 84</td>
</tr>
</tbody>
</table>

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 8. Administrative tasks

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse the different types of parameters.</td>
<td>“Browsing parameters” on page 87</td>
</tr>
<tr>
<td>Copy an existing DB2 entry to the list of DB2 entries on which DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool can be customized.</td>
<td>“Copying DB2 entries” on page 87</td>
</tr>
<tr>
<td>Remove one or more DB2 entries from the associated list.</td>
<td>“Removing DB2 entries” on page 88</td>
</tr>
<tr>
<td>Delete one or more DB2 entries from the master list.</td>
<td>“Deleting DB2 entries” on page 89</td>
</tr>
<tr>
<td>Display a list of customization jobs that have been previously generated.</td>
<td>“Displaying customization jobs” on page 89</td>
</tr>
<tr>
<td>Maintain the customization jobs in the customization library.</td>
<td>“Maintaining customization jobs” on page 90</td>
</tr>
</tbody>
</table>

**Roadmap: Recustomizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool**

This roadmap lists and describes the steps to change parameter values and regenerate customization jobs for DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool after you have customized it for the first time.

The new customization jobs will replace the customization jobs that were previously generated and stored in the customization library. Part of the recustomization process includes selecting or deselecting optional tasks or steps, changing the definitions of parameters that have already been defined, or both. Use the method in this roadmap instead of deleting customization jobs from the customization library.
Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.

Complete the steps in the following table to recustomize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

**Table 9. Required steps for recustomizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specify the product metadata library for the product that you want to recustomize. The name of this library is hlq.SANLDYXIMZDENU.</td>
<td>“Specifying the metadata library for the product to customize” on page 70</td>
</tr>
<tr>
<td>2</td>
<td>Edit the specific tasks, steps, or parameters that need to be changed.</td>
<td>“Defining DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters” on page 77, “Defining LPAR parameters” on page 79, “Defining DB2 parameters” on page 81</td>
</tr>
<tr>
<td>3</td>
<td>Generate the customization jobs for the product or for the DB2 entries on which DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool is ready to be customized.</td>
<td>“Generating customization jobs” on page 83</td>
</tr>
<tr>
<td>4</td>
<td>Submit the new generated customization jobs.</td>
<td>“Submitting customization jobs” on page 84</td>
</tr>
</tbody>
</table>

The following table lists some of the common administrative tasks that you might need to do during the customization process.

**Table 10. Administrative tasks**

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse the different types of parameters.</td>
<td>“Browsing parameters” on page 87</td>
</tr>
<tr>
<td>Copy an existing DB2 entry to the list of DB2 entries on which DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool can be customized.</td>
<td>“Copying DB2 entries” on page 87</td>
</tr>
<tr>
<td>Remove one or more DB2 entries from the associated list.</td>
<td>“Removing DB2 entries” on page 88</td>
</tr>
<tr>
<td>Delete one or more DB2 entries from the master list.</td>
<td>“Deleting DB2 entries” on page 89</td>
</tr>
<tr>
<td>Display a list of customization jobs that have been previously generated.</td>
<td>“Displaying customization jobs” on page 89</td>
</tr>
<tr>
<td>Maintain the customization jobs in the customization library.</td>
<td>“Maintaining customization jobs” on page 90</td>
</tr>
</tbody>
</table>
Specifying the metadata library for the product to customize

You must specify a metadata library for the product that you want to customize.

About this task

The product metadata library contains the information that determines which tasks, steps, and parameters are required to customize DB2 SQL Performance AnalyzerDB2 Autonomic DirectorIMz Sample Tool. This information controls what is displayed on the Product Parameters panel, the LPAR Parameters panel, and the DB2 Parameters panel.

After DB2 SQL Performance AnalyzerDB2 Autonomic DirectorIMz Sample Tool has been SMP/E installed, the default name of the product metadata library is high_level_qualifier.SANLDYXIMZDENU, where high_level_qualifier is all of the segments of the data set name except the lowest-level qualifier.

Procedure

1. Specify option 1 on the Tools Customizer for z/OS panel. The Specify the Metadata Library panel is displayed. This panel contains a list of the product metadata libraries that you specified most recently. If you are using Tools Customizer for the first time, this list is empty, as shown in the following figure:
2. Use one of the following methods to specify the product metadata library:
   - Type the name of a fully qualified partitioned data set (PDS) or an extended partitioned data set (PDSE) in the Metadata library field. Double quotation marks (") cannot be used around the name. Single quotation marks (') can be used but are not required. If you are customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool for the first time, you must use this method.
   - Place the cursor on the library name in the Recent Metadata Libraries list, and press Enter.

Results

If you are customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool for the first time, the Run Discover EXEC panel is displayed. Otherwise, the Customizer Workplace panel is displayed.

What to do next

- Complete the steps that correspond to your environment:

  **Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool for the first time**
  Do not run the DB2 SQL Performance AnalyzerDB2 Autonomics
DirectorIMz Sample Tool Discover EXEC. Press End. The Customizer Workplace panel is displayed. If your environment requires associated DB2 entries, ensure that they are created and associated. If your environment does not require associated DB2 entries, skip this step, and edit DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters.

Customizing DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool from a previous or current customization

Press Enter to run the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC. The Discover Customized Product Information panel is displayed. Specify the required information for running the EXEC.

Discovering DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool information automatically

You can use the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC to discover information from a previous or current customization of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

About this task

Tip: Using the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC to discover information from a previous or current customization saves time and reduces errors that can occur when parameters are specified manually.

DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool provides the Discover EXEC that you will run. Therefore, the information that can be discovered depends on DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

Parameter values that are discovered and parameter values that are specified manually are saved in the data store. If parameter values for the product that you want to customize exist in the data store, Tools Customizer issues a warning before existing values are replaced.

Procedure

1. On the Customizer Workplace panel, issue the DISCOVER command. If you chose to run the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC on the pop-up panel after you specified the product to customize, skip this step.

Tip: You can run any Tools Customizer primary command by using either of the following methods:
   - Place the cursor on the name of the primary command, and press Enter.
   - Type the primary command name in the command line, and press Enter.

The Discover Customized Product Information panel is displayed, as shown in the following figure:
2. Either accept the default values for the following input fields that Tools Customizer generates, or replace the default values with your own values:

**Discover EXEC library (CLIST library)**
- The fully qualified data set name that contains the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC.

**Discover EXEC name**
- The name of the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC.

**Discover output data set**
- The fully qualified data set where output from the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC is stored.

3. Either accept or change the default values in the **Information for Discover EXEC** fields. These fields are generated by DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool. They show the information that is required to run the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC.

4. Issue the **RUN** command to run the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC. Alternatively, save your information without running the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC by issuing the **SAVE** command. If you issue the **RUN** command to run the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC, the parameter information is discovered for DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, and the Customizer Workplace panel is displayed.

**Results**

The discovered parameter values for DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool replace any existing values.

**What to do next**

The next step depends on your environment:
• If DB2 entries were not discovered, or if you need to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool on new DB2 entries, create and associate the entries.

• If DB2 entries were discovered and you want to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool on only these entries, define the parameters.

Related tasks:

“Creating and associating DB2 entries”
You can create new DB2 entries and associate them with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

“Defining parameters” on page 77
To customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, you must define DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters, LPAR parameters, and DB2 parameters, if your customization requires DB2 entries.

Creating and associating DB2 entries
You can create new DB2 entries and associate them with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

About this task
The list of associated DB2 entries is on the Customizer Workplace panel.

Procedure
1. Issue the ASSOCIATE command on the Customizer Workplace panel. The Associate DB2 Entry for Product panel is displayed, as shown in the following figure:
2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.
   a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

   **Figure 8. The Associate DB2 Entry for Product panel**

   2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.
      a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

      **Figure 8. The Associate DB2 Entry for Product panel**

      2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.
         a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

         **Figure 8. The Associate DB2 Entry for Product panel**

         2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.
            a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

            **Figure 8. The Associate DB2 Entry for Product panel**

            2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.
               a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

               **Figure 8. The Associate DB2 Entry for Product panel**

               2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.
                  a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

                  **Figure 8. The Associate DB2 Entry for Product panel**

                  2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.
                     a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

                     **Figure 8. The Associate DB2 Entry for Product panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

                     **Figure 9. The Create a DB2 Entry panel**

                     b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or both for the new DB2 entry. Press Enter to continue or End to cancel. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

                     **Figure 9. The Create a DB2 Entry panel**
To insert multiple DB2 entries, specify the `lnn` line command, where `lnn` is the number of DB2 entries to be inserted.

You will define specific parameters for these new DB2 entries on the DB2 Parameters panel. This panel is displayed after you select these new DB2 entries and issue the line command to generate the jobs, after you issue the primary command to generate the jobs for all associated DB2 entries, or when you manually edit the DB2 parameters.

The Associate DB2 Entry for Product panel is displayed, and the new DB2 entry is displayed in the master list, as shown in the following figure:

```
CCQPDAD	Associate DB2 Entry for Product	Row 1 to 1 of 1
Command ===>	Scroll ===>	CSR
Select any of the following DB2 entries to add them to the Customizer Workplace panel. You use the Customizer Workplace panel to choose the DB2 subsystems, data sharing members, and group attach names on which to customize the product.
Commands: CREATE - Create a new DB2 entry
Product to Customize
  Product metadata library : ANL.ALIAS.SANLDENU > LPAR . . : MVS1
  Product name . . . . . . . : DB2 SQL Performance Analyzer
  Product version . . . . . : 4.2.0
Line commands: A - Associate  C - Copy
Cmd SSID GrpAttach
DB22 --
------------------------------- End of DB2 entries -----------------------------
```

Figure 10. The Associate DB2 Entry for Product panel with a new DB2 entry in the master list

c. Repeat steps b and c for each DB2 entry that you want to create.
d. When you have created all the DB2 entries, associate them with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, or press End to display the Customizer Workplace panel.

3. Associate DB2 entries.
   a. Specify `A` against one or more DB2 entries in the master list, and press Enter to associate them with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

Results

The Customizer Workplace panel is displayed with the associated DB2 entries displayed in the associated list.

What to do next

Define the parameters.

Related concepts:

“Tools Customizer terminology” on page 257

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.
Defining parameters

To customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, you must define DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters, LPAR parameters, and DB2 parameters, if your customization requires DB2 entries.

About this task

You must define the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters first for the following reasons:

- If you ran the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC, you must review the values that were discovered.
- If you select optional tasks and steps on the Product Parameters panel that affect the DB2 entry on which you will customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, additional parameters might be displayed on the DB2 Parameters panel.
- If other steps must be completed in a specific sequence, customization notes on the Product Parameters panel will display the correct sequence.

Defining DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters

DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters are specific to DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

About this task

If you ran the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC, you must review the parameters that were discovered.

Procedure

1. Specify E next to the Product parameters field on the Customizer Workplace panel, and press Enter. The Product Parameters panel is displayed, as shown in the following figure. If other steps must be completed in a specific sequence before you define the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters, a note labeled Important will display the correct sequence on this panel.
Complete the following tasks to customize the products. The required tasks and steps are preselected. Ensure that all parameters are specified for each selected step within a task. Press End to save and exit.

Commands: SAVE - Save parameter values
Line Commands: / - Select

Required parameters
HLQ of SQL PA installation ........ ANL.ANL420

/ Create files needed for TSO and batch execution
- Drop DB2 objects
- Prefix for output reports and work files
- High-level qualifier of DB2 Path Checker
- Create generic plan tables
- Create new DSN3OATH
- Create new DSN30SGN

Figure 11. The Product Parameters panel
2. Select any required tasks and steps, and specify values for any parameters. After you select a task or step with a slash (/), put the cursor in the selected field and press Enter. If tasks, steps, and parameters are required, they are preselected with a slash (/). Otherwise, they are not preselected. All of the required parameters have default values, which you can either accept or change.

Tips:
- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
- For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
- The following elements apply to specific fields:
  - **Add...** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add...**, and press Enter. Use the displayed panel to add or delete additional values.
  - **List...** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List...**, and press F1 or the key that is mapped to Help.
  - **More...** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More...**, and press Enter.

3. Optional: Select other tasks and steps with a slash (/) and press Enter to activate the input fields. Either accept or change the default values that are displayed.

4. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the Product Parameters panel.

**Results**

The Customizer Workplace panel is displayed, and the status of the product parameters is Ready to Customize.

**What to do next**

If the status of other parameters on the Customizer Workplace panel is Incomplete or Discovered, edit these parameters.

**Related tasks:**
- “Defining LPAR parameters”
  LPAR parameters are parameters on the local LPAR that are required to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.
- “Defining DB2 parameters” on page 81
  DB2 parameters are parameters for a DB2 entry.

**Defining LPAR parameters**

LPAR parameters are parameters on the local LPAR that are required to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.
Procedure

1. Specify E next to the **LPAR parameters** field, and press Enter. The LPAR Parameters panel is displayed, as shown in the following figure:

   ![Figure 12. The LPAR Parameters panel](image)

2. Specify values for all required parameters that are displayed. Many parameters have default values, which you can either accept or change.

   **Tips:**
   - In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
   - For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
   - The following elements apply to specific fields:
     - **Add...** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add...**, and press Enter. Use the displayed panel to add or delete additional values.
     - **List...** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List...**, and press F1 or the key that is mapped to Help.
     - **More...** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More...**, and press Enter.

   The following LPAR parameters can contain 1 - 64 values:
• LPAR macro library
• Message library
• Panel library
• Skeleton library
• ISPF table input library
• ISPF user profile library
• File tailoring output library
• Link list library
• Command procedures library
• Macro library
• Link-edit library
• Load library
• Started task library name

3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the same panel.

Results

The Customizer Workplace panel is displayed, and the status of the LPAR parameters is Ready to Customize.

What to do next

If the status of other parameters on the Customizer Workplace panel is Incomplete or Discovered, edit these parameters.

Related tasks:
“Defining DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters” on page 77
DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters are specific to DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.
“Defining DB2 parameters”
DB2 parameters are parameters for a DB2 entry.

Defining DB2 parameters

DB2 parameters are parameters for a DB2 entry.

About this task

If you did not run the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC, you must create and associate one or more DB2 entries before you can define the DB2 parameters. For more information, see “Creating and associating DB2 entries” on page 74.

Procedure

1. Specify E next to one or more DB2 entries in the associated list, which is in the Associated DB2 Entries and Parameter Status section on the Customizer Workplace panel, and press Enter. The DB2 Parameters panel is displayed, as shown in the following figure:
Enter values for all of the DB2 parameters. Press End to save and exit.

Commands: SAVE - Save parameter values

Product to Customize
Product metadata library : ANL.ALIAS.SANLDENU > LPAR...: MVS1
Product name...........: DB2 SQL Performance Analyzer > Version...: 4.2.0

More: +

DB2 subsystem ID...........: DB01
Group attach name...........

General DB2 Information
Mode.....................: NFM (CM,CM8,CM9,NFM)
Level number...............: 101 (910,101,111)

DB2 Libraries
Load Library.............. DSN.SDSNLOAD > Add...
Run Library.............. DSN.SDSNRUNLIB > Add...
Exit Library.............. DSN.SDSNEXIT > Add...
Sample library............ DSN.SDSSAMPLE > Add...
Macro library............. DSN.SDSNMACS > Add...

DB2 Buffer pools
Name of the 4 KB bufferpool..... BP0
Name of the 8 KB bufferpool..... BP8K0

DB2 Utilities
Plan name for the DSNTIAD utility.....

SQL PA DB2 Objects
Database name............. ANLDBASE
Storage group name......... SYSDEFLT

Figure 13. The DB2 Parameters panel
2. Specify values for all parameters that are displayed.

Tips:

- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
- For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
- The following elements apply to specific fields:
  - **Add...** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add...**, and press Enter. Use the displayed panel to add or delete additional values.
  - **List...** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List...**, and press F1 or the key that is mapped to Help.
  - **More...** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More...**, and press Enter.

Many parameters have default values, which you can either accept or change.

3. Press End to save your changes and exit, or issue the **SAVE** command to save your changes and stay on the same panel.

Results

The status of the DB2 entries that you selected on the Customizer Workplace panel is Ready to Customize.

What to do next

If the status of other parameters on the Customizer Workplace panel is Incomplete or Discovered, edit these parameters.

Related tasks:

- “Defining DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters” on page 77
- DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters are specific to DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.
- “Defining LPAR parameters” on page 79
- LPAR parameters are parameters on the local LPAR that are required to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

Generating customization jobs

To generate customization jobs for DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool and any associated DB2 entries, issue the **GENERATEALL** command, or select one or more DB2 entries on which to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

Procedure

Generate the customization jobs by using one of the following methods.
• If you want to generate customization jobs at the product level and for any associated DB2 entries, issue the GENERATEALL command, and press Enter.

• If you want to generate customization jobs for specific DB2 entries, select the DB2 entries by specifying the G line command against them, and press Enter. The available DB2 entries are in the associated list in the Associated DB2 Entries and Parameter Status section.

Important: Regenerating customization jobs will replace any existing jobs, including jobs that you might have manually modified after they were generated.

Results

If the status is Incomplete or Discovered for DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool parameters, LPAR parameters, or DB2 parameters, Tools Customizer automatically starts an editing session for the types of parameters that are required. The session continues until the panel for each type of required parameter has been displayed.

What to do next

If an automatic editing session is started, accept the displayed parameter values or define values for the required types of parameters, select optional parameters, tasks, or steps for your environment, and save the parameter values. Otherwise, the customization jobs are generated, and you can submit them.

Tip: If the customization jobs are generated, but you are not ready to submit them, you can see them later by issuing the JOBLIST command on the Customizer Workplace panel. The JOBLIST command displays the Finish Product Customization panel, which you can use to submit the jobs.

Submitting customization jobs

Submit the customization jobs to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

Before you begin

Ensure that the correct jobs are generated.

About this task

The following figure shows part of the Finish Product Customization panel. The table on this panel shows the customization jobs that are generated by Tools Customizer. They are grouped by job sequence number.
The member-naming conventions depend on whether the customization jobs are for DB2 entries, and LPAR, or the product.

**Customization jobs for DB2 entries**

The members use the following naming convention:

\[
<job_sequence_number><job_ID><DB2_entry_ID>
\]

where

- **job_sequence_number**
  Two alphanumeric characters, A0 - Z9, that Tools Customizer assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

- **job_ID**
  Characters 4 - 7 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool assigns the template name.

---

**CCQPCST** Finish Product Customization Row 1 to 23 of 41

Submit the members in the order in which they apply to each DB2 entry. To submit the job, edit the member and issue the TSO SUBMIT command, or edit the customized library and submit the jobs from there.

**Product to Customize**

Product metadata library : ANL.ALIAS.SANLDENU > LPAR ... : MVS1

Product name . . . . . . : DB2 SQL Performance Analyzer > Version . . : 4.2.0

Line Commands: E - Edit  B - Browse

Product customization library : ANL.PRODUCT.CUST.$MVS1$.ANL42Q >

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Member</th>
<th>SSID</th>
<th>GrpAttach</th>
<th>Template</th>
<th>Generated Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0DROPAA DB01 -- ANLDROPD 2013/06/25 Drops DB2 objects from a previous customization.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1DCREAA DB01 -- ANLDCRE8 2013/06/25 Creates the necessary DB2 objects.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2PCREAA DB01 -- ANLPCRE8 2013/06/25 Creates all required SQL Performance Analyzer objects.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3ICREAA DB01 -- ANLICRE8 2013/06/25 Creates ssidCNTL, ssidPARM, and ssidVAR.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4JCREAA DB01 -- ANLJCRE8 2013/06/25 Creates the JCL that is stored in the library.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5BNDGAA DB01 -- ANLBNDGR 2013/06/25 Binds the packages and plans, and provides access to the DB2 objects.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1QMONC DB01 -- ANLQMFGV 2013/06/25 Creates the CLIST that is used for the customization.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**CCQPCST** Finish Product Customization Row 1 to 6 of 6

For a first-time customization, submit the jobs in the members in the order in which they apply to the DB2 entries. Otherwise, submit only the necessary jobs that were generated after changes were made. To submit jobs, browse the members and issue the TSO SUBMIT command.

Line Commands: E - Edit  B - Browse

Product customization library: SAGREGG.TCZ.CUSTLIBS.TOOLS.$MVS1$.DYX150

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Member</th>
<th>SSID</th>
<th>GrpAttach</th>
<th>Template</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0DROPAA DB01 -- DYXDROP 2014/07/31 Drop DB2 Autonomics Director packages.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1DOLIAA DB01 -- DYXDOLI 2014/07/31 Create DB2 Autonomics Director objects.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2FREAA DB01 -- DYXFREE 2014/07/31 Free packages and plans.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3BINDAA DB01 -- DYXBIND 2014/07/31 Binds packages and plans.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4GRANAA DB01 -- DYXGRANT 2014/07/31 Grants EXECUTE authority.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5WLMAA DB01 -- DYXWLM 2014/07/31 Autonomics WLM address space.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Figure 14. The Finish Product Customization panel
DB2_entry_ID
Two alphanumeric characters, AA - 99, that Tools Customizer assigns to a DB2 entry.

For example, the XYZBNDDB2_entry_ID_1 and XYZBNDDB2_entry_ID_2 jobs are generated from the XYZBNDGR template, and the XYZ4DB2_entry_ID_1 and XYZ4DB2_entry_ID_2 jobs are generated from the XYZ4 template. If the jobs are generated on two DB2 entries, the following member names are listed sequentially: A0BNDGAA, A0BNDGAB, A14AA, A14AB.

Customization jobs for an LPAR or the product
The members use the following naming convention:
<job_sequence_number><job_ID>

where

job_sequence_number
Two alphanumeric characters, A0 - Z9, that Tools Customizer assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

job_ID
Characters 4 - 8 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. For example, for the XYZMAKE template, the job ID is MAKE. For the XYZ4 template, the job ID is M. DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool assigns the template name, and it is displayed in the Template column.

For example, the XYZBNDGR job is generated from the XYZBNDGR template, and the XYZ4 job is generated from the XYZ4 template. The following member names are listed sequentially: A0BNDGAR, A14.

Procedure
1. Submit the generated customization jobs by following the process that you use in your environment or by using the following method:
   a. Specify B against a customization job or the product customization library, and press Enter. An ISPF browsing session is started.
   b. Browse the customization job or each member in the library to ensure that the information is correct.
   c. Run the TSO SUBMIT command.
2. Press End.

Results
DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool is customized, and the Customizer Workplace panel is displayed. The status is Customized for the DB2 entries on which DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool was customized.

What to do next
You can generate more customization jobs for other DB2 entries, view a list of customization jobs that you previously generated, or recustomize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorMz Sample Tool.
Browsing parameters
You can browse the product parameters, the LPAR parameters, and the DB2 parameters in read-only mode.

Procedure
1. On the Customizer Workplace panel, specify B next to the Product parameters field, the LPAR parameters field, or the DB2 entry that you want to browse, and press Enter. The panel that corresponds to your specification is displayed.
2. Press End to exit.

Copying DB2 entries
You can copy associated and not associated DB2 entries to other DB2 entries or to new DB2 entries.

About this task
Go to the step that applies to your environment:
• To copy an associated DB2 entry to another associated DB2 entry or to an entry that is not associated, go to step 1.
• To copy an associated DB2 entry to a new entry, go to step 2.
• To copy a DB2 entry that is not associated to a new entry, go to step 3.

Procedure
1. To copy an associated DB2 entry to another associated DB2 entry or to an entry that is not associated, complete the following steps:
   a. Specify C against a DB2 entry in the associated list of DB2 entries on the Customizer Workplace panel, and press Enter. The Copy Associated DB2 Entry panel is displayed.
   b. Select one or more DB2 entries to which information will be copied by specifying the / line command, and press Enter. The Associated column indicates whether the DB2 entry is associated.

   Tip: To copy information into all of the DB2 Entries in the list, issue the SELECTALL primary command, and press Enter. The Copy DB2 Parameter Values panel is displayed.

   c. Specify an option for copying common and product-specific DB2 parameter values. Common DB2 parameter values apply to all DB2 entries for all products that you have customized by using Tools Customizer. Product-specific DB2 parameter values apply only to the product that you are currently customizing.
      • To copy the common DB2 parameter values and the product-specific DB2 parameter values, specify option 1, and press Enter.
      • To copy only the product-specified DB2 parameter values, specify option 2, and press Enter.

   In some cases, the DB2 parameter values might contain the DB2 subsystem ID as an isolated qualifier in data set names. For example, in the DB01.DB01TEST.DB01.SANLLOAD, data set name, the DB01 subsystem ID is isolated in the first and third qualifiers but is not isolated in the second qualifier. When the DB2 subsystem ID is an isolated qualifier in data set names, the Change DB2 Subsystem ID in DB2 Parameter Values panel is displayed. Otherwise, the Customizer Workplace panel is displayed.
d. If the Change DB2 Subsystem ID in DB2 Parameter Values panel is displayed, specify an option for changing the subsystem IDs. Otherwise, skip this step.
   - To change the subsystem ID in isolated qualifiers in data set names, specify option 1, and press Enter.
   - To use the same subsystem ID in all values, specify option 2, and press Enter.

The Customizer Workplace panel is displayed with the copied associated entry in the list.

2. To copy an associated DB2 entry to a new entry, complete the following steps:
   a. Specify C against a DB2 entry in the associated list of DB2 entries on the Customizer Workplace panel, and press Enter. The Copy Associated DB2 Entry panel is displayed.
   b. Issue the CREATE command. The Create DB2 Entries panel is displayed.
   c. Specify the SSID, the group attach name, or both in the appropriate columns for each new DB2 entry, and press Enter.

   Tip: To add rows for additional entries, specify the Insert line command, where nn is the number of entries to be created, and press Enter. The Copy Associated DB2 Entry panel is displayed with the new entries in the list. The new entries are preselected.
   d. Press Enter to complete the copy process. The Customizer Workplace panel is displayed with the copied entries in the list.

3. To copy a DB2 entry that is not associated to a new entry, complete the following steps:
   a. Issue the ASSOCIATE command on the Customizer Workplace panel. The Associate DB2 Entry for Product panel is displayed.
   b. Select one or more DB2 entries by specifying the / line command, and press Enter. The Copy a DB2 Entry panel is displayed.
   c. Specify the SSID, the group attach name, or both in the appropriate columns for the new DB2 entry, and press Enter. The Associate DB2 Entry for product panel is displayed with the copied entry in the list.
   d. If you want to associate the copied entry, specify A against it, and press Enter. The Customizer Workplace panel is displayed with the copied entries in the list.

What to do next

Edit any of the parameters or generate the jobs.

Related concepts:
“Tools Customizer terminology” on page 257
Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Removing DB2 entries

You can remove DB2 entries from the associated list.
About this task

When you remove DB2 entries from the associated list, any customization jobs for the entries are removed from the list of jobs on the Finish Product Customization panel, and they are deleted.

Procedure

On the Customizer Workplace panel, specify R next to one or more DB2 entries that you want to remove, and press Enter. The selected DB2 entries are removed from the associated list and added to the master list on the Associate DB2 Entry for Product panel, and the customization jobs are deleted.

Related concepts:
“Tools Customizer terminology” on page 257

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Deleting DB2 entries

You can delete DB2 entries from the master list.

About this task

When you delete DB2 entries from the master list, any associations and all customization jobs for products that are customized on the entries will be deleted.

Procedure

1. On the Customizer Workplace panel, issue the ASSOCIATE command. The Associate DB2 Entry for Product panel is displayed.
2. Specify D next to one or more DB2 entries that you want to delete, and press Enter. If the entry is associated with any products, the Delete Associated DB2 Entry panel for the first DB2 entry that you selected is displayed. Otherwise, the Delete DB2 Entry panel is displayed.
3. To delete the DB2 entries, press Enter. If the DB2 entries are associated with any products in the table on the Delete Associated DB2 Entry panel, any associations and all customization jobs for the products that are customized on it are deleted. Otherwise, only the DB2 entries are deleted. If you selected multiple DB2 entries to delete, the next DB2 entry that you selected is displayed on either the Delete Associated DB2 Entry panel or the Delete DB2 Entry panel. Otherwise, the Associate DB2 Entry for Product panel is displayed.

What to do next

If you selected multiple DB2 entries to delete, repeat step 3 until all selected entries are deleted. Then, continue the customization process.

Displaying customization jobs

You can view a list of the members that contain the customization jobs before or after you submit the jobs.

About this task

The customization jobs that you generate for one DB2 entry are also displayed when you customize DB2 SQL Performance Analyzer, DB2 Autonomics Director, IBMz Sample Tool for another DB2 entry later.
**Procedure**

On the Customizer Workplace panel, issue the JOBLIST command. The Finish Product Customization panel is displayed. This panel shows the list of jobs that you have previously generated. They are grouped by job sequence number. Use this panel to browse or edit the generated jobs before you submit them.

**Maintaining customization jobs**

Instead of deleting customization jobs outside of Tools Customizer, you can maintain the correct jobs for DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool by completing the steps for recustomization.

**About this task**

You cannot delete or rename customization jobs from the customization library by starting an ISPF browse or edit session from the Finish Product Customization panel. If you try to delete customization jobs by using this method, the CCQC034S message is issued. If you try to rename customization jobs, the CCQC035S message is issued.

If you delete or rename customization jobs from the customization library by using ISPF outside of Tools Customizer, Tools Customizer will not recognize that the jobs were deleted, and the Finish Product Customization panel will still display them. If you browse or edit jobs that were deleted from the library outside of Tools Customizer, the CCQC027S message is issued.

**Procedure**

To maintain the correct customization jobs in the customization library, complete the steps for recustomization.

**Using Tools Customizer in a multiple-LPAR environment**

Currently, Tools Customizer supports only the local LPAR; however, you can propagate customizations to additional LPARs by using either of two different methods.

**About this task**

In a multiple-LPAR environment, Tools Customizer identifies the LPAR to which you are logged on. Tools Customizer uses this LPAR name for several different parameter settings, one of which is the data store. When you use the data store during the customization of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool that is on a different LPAR, Tools Customizer issues message CCQD586S, which indicates that the product has already been customized based on values from the data store on the first LPAR. This message is issued to prevent the data store from becoming corrupted.

This behavior occurs in the following conditions:

- Tools Customizer is installed on a DASD device that is shared by multiple LPARs.
- After a product is customized by using Tools Customizer, the data store is copied to another LPAR.
Procedure

To customize products running against a DB2 subsystem on an LPAR where Tools Customizer is not installed, consider using one of the following methods:

Install one instance of Tools Customizer on one LPAR
If you intend to reuse the customization values for all the instances of your products on all LPARs, use this method.
1. Associate all the DB2 entries in this one instance of Tools Customizer. The LPARs on which the DB2 subsystems reside do not matter.
2. Generate the customization jobs for each DB2 entry.
3. Copy the generated customization jobs to the LPAR to run against the specific DB2 entries. Some LPAR-specific edits might be required. You can make these edits in the customized jobs that you copied. Note that this situation is one of the few situations where you might need to make manual changes to the jobs that are customized by Tools Customizer.

Install one instance of Tools Customizer on each LPAR
If you do not want to reuse previous customization values and you want to start new customizations, use this method.

Important: This method will likely not be the preferred approach for most organizations because most organizations tend to use similar or identical customization values for each product instance on all LPARs.

Optional: Creating generic plan tables

Optionally, you can create generic plan tables by using Tools Customizer. Tools Customizer will create the necessary JCL, but you must manually complete some steps before and after you submit the customization job.

Before you begin

Ensure that you selected the optional task to create generic plan tables on the Product Parameters panel in Tools Customizer and that Tools Customizer generated the customization job.

About this task

Before you can create generic plan tables, you must create secondary authorization identifiers (authorization IDs) that will act as the owners of the generic DB2 SQL Performance Analyzer plan tables. These secondary authorization IDs allow DB2 SQL Performance Analyzer to switch any user from their current authorization ID to one of the special DB2 SQL Performance Analyzer secondary authorization IDs, which are shared by all DB2 users.

Before you submit the customization job, you must complete the following steps manually.

Procedure

1. Define the secondary authorization IDs in the DSN3@ATH exit. The sample exit code for the DSN3@ATH exit is distributed with DB2 as member DSN3SATH in the DSNxxx.SDSNSAMP sample library.
a. Obtain the most current copy of DSN3SATH and copy it to hlq.SANLSAMP. Many sites customize this exit by adding their own code to the DB2 sample program.

b. Copy the DB2 SQL Performance Analyzer modifications from hlq.SANLSAMP(ANLAUTH), and paste them into the DSN3@ATH exit immediately after the SATH090 label. The following example shows the code with the secondary authorization IDs after the SATH090 label:

```
SATH090 DS 0H
* ===================================================================*
* APPEND THE ANL SECONDARY AUTHIDS TO THE END OF THE SECONDARY LIST *
* =========================================================================* 
* THERE ARE 245 POSSIBLE ENTRIES ON THE SECONDARY AUTHID LIST. INSERT *
* AS MANY ANL IDS AS WILL FIT ON THE END, WITHOUT EXCEEDING 245 LIMIT *
* THE NUMBER OF ENTRIES AND THE ANL USER NAMES MUST CORRESPOND TO THE *
* ENTRIES IN ANL REGISTRY TABLE. THE NUMBER OF IDS (10) IS ARBITRARY. *
*==================================================================*
LA R4,AIDLSEC R4 ADDRESS OF SECONDARY AREA
L R0,AIDLSCNT R0 NUMBER OF 2NDARY SLOTS USED
LA R3,X'F5' R3 CONSTANT 245 MAXIMUM SLOTS
SR R3,R0 R3 HOLDS REMAINING SLOTS UNUSED
BC 2,ANLOK OK IF >0 RESULT (CONDITION = 2)
B ENDANL ELSE NO SLOTS, CANNOT INSERT IDS
ANLOK MH R0,X'0008' MULTIPLY SLOTS * 8 = BYTES USED
AR R0,R4 R0 STORE STOPPING ADDRESS HERE
ANLSEC CLC 0(R4),=CL8' ' IS THIS A BLANK SLOT?
BE ADDANL1 GOOD, ADD SQL/PA SECONDARIES
LA R4,8(R4) BUMP UP AND CONTINUE
CR R4,R0 HAVE WE EXHAUSTED THE LIST?
BNL ENDANL YES, NO MORE ROOM AVAILABLE
B ANLSEC NO, TRY NEXT SLOT
ADDANL1 DS 0H
MVC 0(R4),=CL8'ANLUSER1' ADD SECONDARY ID 1
LA R4,8(R4) BUMP TO NEXT
BCT R3,ADDANL2 YES, ROOM FOR ANOTHER
B ENDANL NO, WE ARE DONE
ADDANL2 MVC 0(R4),=CL8'ANLUSER2' ADD SECONDARY ID 2
LA R4,8(R4) BUMP TO NEXT
BCT R3,ADDANL3 YES, ROOM FOR ANOTHER
B ENDANL NO, WE ARE DONE
ADDANL3 MVC 0(R4),=CL8'ANLUSER3' ADD SECONDARY ID 3
LA R4,8(R4) BUMP TO NEXT
BCT R3,ADDANL4 YES, ROOM FOR ANOTHER
B ENDANL NO, WE ARE DONE
ADDANL4 MVC 0(R4),=CL8'ANLUSER4' ADD SECONDARY ID 4
LA R4,8(R4) BUMP TO NEXT
BCT R3,ADDANL5 YES, ROOM FOR ANOTHER
B ENDANL NO, WE ARE DONE
ADDANL5 MVC 0(R4),=CL8'ANLUSER5' ADD SECONDARY ID 5
LA R4,8(R4) BUMP TO NEXT
BCT R3,ADDANL6 YES, ROOM FOR ANOTHER
B ENDANL NO, WE ARE DONE
ADDANL6 MVC 0(R4),=CL8'ANLUSER6' ADD SECONDARY ID 6
LA R4,8(R4) BUMP TO NEXT
BCT R3,ADDANL7 YES, ROOM FOR ANOTHER
B ENDANL NO, WE ARE DONE
ADDANL7 MVC 0(R4),=CL8'ANLUSER7' ADD SECONDARY ID 7
LA R4,8(R4) BUMP TO NEXT
BCT R3,ADDANL8 YES, ROOM FOR ANOTHER
B ENDANL NO, WE ARE DONE
ADDANL8 MVC 0(R4),=CL8'ANLUSER8' ADD SECONDARY ID 8
LA R4,8(R4) BUMP TO NEXT
BCT R3,ADDANL9 YES, ROOM FOR ANOTHER
B ENDANL NO, WE ARE DONE
ADDANL9 MVC 0(R4),=CL8'ANLUSER9' ADD SECONDARY ID 9
LA R4,8(R4) BUMP TO NEXT
```
Optional: Define the secondary authorization IDs in the DSN3@SGN exit. Consider modifying the DSN3@SGN exit with the same set of IDs that you added to the DSN3@ATH exit. Even though the DSN3@SGN exit is used only for IMS and CICS sign-on authorizations, keeping these two exits consistent is a best practice. This step is identical to step 1 except that you add the secondary authorization IDs to the DSN3@SGN exit immediately after the SSGN090 label.

3. Submit the customization job.

4. Test the modified DSN3@ATH and DSN3@SGN authorization exits in isolation before putting the exits into production. To facilitate testing in isolation, load the DSN3@ATH and DSN3@SGN exits into the SYS1.DSNEXIT library or a similar stand-alone library so that they can be copied into SYS1.DSNLINK or SYS1.DSNLOAD during a test window and verified.
   a. Assemble and link the modules into SYS1.DSNEXIT or an equivalent library that does not currently contain the production exits.
      
      **Attention:** If the library is already on the link list, users have immediate access.
   
   b. During a test window or on a test system, rename the current exits (if they exist) and copy the new modules into SYS1.DSNLINK or SYS1.DSNLOAD.
   
   c. From a TSO session, enter the DB2I command. Select an empty file for input and enter the following DB2 statement: `SET CURRENT SQLID = 'ANLUSER1'`. If the authorization exits have been correctly modified, an SQLCODE of 0 is returned. If no valid secondary authorization ID by that name exists, DB2 returns an SQLCODE of -553, which means that the exit modifications are not working. If additional coding errors are present, the exits might abend. If you encounter a problem, correct the error and retest the exits after you rename the old production exits to their proper names.
   
   d. If the SQLCODE is 0, set all of the SQL PA secondary authorization IDs that you defined. This step verifies that all the secondary authorization IDs are spelled correctly.
   
   e. Migrate the DSN3@ATH and DSN3@SGN modules to a production library on the link list (refer to SYS1.PARMLIB member LNKLSTxx), such as SYS1.DSNLINK or SYS1.DSNLOAD, for access by all users.

**Optional: Interfacing with DB2 Query Monitor**

You can use an optional task in Tools Customizer to set up the interface between DB2 SQL Performance Analyzer and DB2 Query Monitor. If you are also installing DB2 Query Monitor, you might want to enable this interface.

**About this task**

This optional configuration task allows DB2 Query Monitor to call DB2 SQL Performance Analyzer for further analysis of problem SQL.

Tools Customizer creates the ANLCIQM CLIST in the `high_level_qualifier.SANLCLST` library for interfacing DB2 SQL Performance Analyzer with the DB2 Query Monitor.
Procedure

To enable DB2 SQL Performance Analyzer to interface with DB2 Query Monitor:

1. Select and specify values for all of the required Set up interface to Query Monitor task parameters. The ANLCIQM CLIST is created for you.
2. Set QDEBUG to NO. The remaining parameters are set by DB2 Query Monitor upon invocation.
3. In DB2 Query Monitor, export SQL text of interest to a data set through the Export SQL Text to DSN panel, shown in Figure 15.
4. Specify Y in the Execute SQL PA against exported data set field, to invoke DB2 SQL Performance Analyzer against the data set where you are exporting SQL text. Access the Export SQL Text to DSN panel when you view operational summaries of activity, exceptional query activity, or in-process activity.

Example

Optional: Configuring DB2 SQL Performance Analyzer for use with QMF

You can optionally configure DB2 SQL Performance Analyzer to interface with QMF. Tools Customizer creates the necessary JCL, but some of the following steps must be completed before running the JCL.

Procedure

1. Create the interface between DB2 SQL Performance Analyzer and QMF.
   a. Copy the contents of the ANLQM member in the high_level_qualifier.SANLSAMP library.
   b. In the DSQUEG1 member, paste the ANLQM code between the code that is labeled GET ADDRESSES OF DXEXCBA AND DXEGOVA CNTL BLOCKS, which is located before the USING WORK,WORKPTR statement, and the code that is labeled SET SCOPE VALUE TO ZERO ON INITIALIZATION..., which is located before the CLI GOVFUNCT,GOVINIT statement, in the governor exit program in any supported version of QMF.
   c. Save the changes to DSQUEG1.
2. Customize the interface between DB2 SQL Performance Analyzer and QMF by modifying the parameters that define the target host system.
   a. Open the DB2 SQL Performance Analyzer Governor Interface program, ANLGOV1, which is located in the high_level_qualifier.SANLSAMP library, and find the ANLPARM parameters list near the end of the program code. The following code shows the default ANLPARM settings in ANLGOV1:

   ```
   CQM$EXPT--------------- Export SQL Text to DSN ---------------
   Option ==> _______________________________________________________
   Export to data set ... _____________________________________________
   Member. ________ (Required if data set is a PDS)
   Execute SQL/PA against exported data set Y/N Y
   Press Enter to process request or PF3/CANCEL to exit
   ```

   Figure 15. Query Monitor panel to invoke DB2 SQL Performance Analyzer against an exported data set containing SQL
**ANLPARM** DC H'800' STORE UP TO SQL/PA PARAMETERS HERE

**ANLCARDS** DS 0CL800 PASSED TO PLI PROGRAM ANLQMF

- **DC CL16**'SUBSYST DSN ' REQUIRED SUBSYSTEM ID FOR CAF
- **DC CL16**'REPORTS MAX' MAX= WARNINGS ONLY; ANY=ALL QUERIES
- **DC CL16**'VERSION V10R1' CURRENT DB2 RELEASE V9R1 V10R1 ...
- **DC CL16**'SUBVERS V10NFM' DB2 LEVEL IS V9COM V9NFM V10CM...
- **DC CL16**'STORAGE 3390-3' DASD POOL IS 3390 MOD 3
- **DC CL16**'BUFFHIT 000' ALL I/O IS PHYSICAL (BP HIT RATIO)
- **DC CL16**'DEGREES ANY' ENABLE PARALLEL I/O: ANY OR ONE|1
- **DC CL16**'REFRESH ANY' REFRESH MQTS USE FOR APS: ANY|NO
- **DC CL16**'CONNECT QMF' REQUIRED CONNECTION = QMF ALWAYS
- **DC CL16**'QUALIFY ' DEFAULT HIGH LEVEL QUALIFIER
- **USEQUAL** DC CL16'SETPLAN YES' ALLOW USEPLAN TO FUNCTION
- **DC CL16**'ENGINES 00001' TOTAL ENGINES ON BOX
- **DC CL16**'LPARENG 01.000' PORTION OF BOX FOR THIS IMAGE
- **DC CL16**'ESASORT YES ' YES HARDWARE SORT ENABLED
- **DC CL16**'ESCAMP YES ' YES HARDWARE DATA COMPRESSION
- **DC CL16**'BUFFERS 00020000' 4K APPLICATION BUFFER POOL
- **DC CL16**'BUFFBK 00020000' 8K APPLICATION BUFFER POOL
- **DC CL16**'BUFF16K 00010000' 16K APPLICATION BUFFER POOL
- **DC CL16**'BUFF32K 00001000' 32k APPLICATION BUFFER POOL
- **DC CL16**'CPUTIME 00010' WARNINGS FOR 10 SECs CPU TIME
- **DC CL16**'ELAPSED 00120' 2 MINUTES ELAPSED TIME NOTICE
- **DC CL16**'IOCALS 00010000' 1000 I/O ISSUE A WARNING
- **DC CL16**'COSTING 0000015' FLAG ANY QUERY > $15
- **DC CL16**'COSTQUN 001' FLAG ANY QUERY > 100 QUNITS
- **DC CL16**'CPUCOST 0800.000' CPU TIME AT $800 PER HOUR
- **DC CL16**'IOSCOST 0010.000' $10 PER 1000 I/O (PHYSICAL) (P#30)
- **DC CL16**'TIMCOST 0010.000' $10 PER HOUR CONNECT TIME
- **DC CL16**'MONEYIS DOLLARS ' MONETARY UNITS
- **DC CL16**'CURRSYM $' CURRENCY SYMBOL
- **DC CL16**'PRECISE YES ' EXTRA DB2 OPTIMIZER COSTS
- **DC CL16**'DBTRACE NO ' TRACE FOR QMF TEST: ON|ALL|DMP|NO
- **DC CL224** ' RESERVATIONS (+224 FOR 50 PARMS)

* ******************************************************
* SOME SITES MAY WISH TO ALLOW USERS TO SPECIFY THE HLQ OF THE PLAN_  
* TABLE, IN LIEU OF SELECTION FROM REGISTRY: 'SETPLAN YES' IS REQUIRED  
* TO ENABLE THIS FEATURE. THEN, THE CURRENT SQLID, TAKEN FROM THE QMF_  
* CONTROL BLOCK, IS INSERTED INTO THE 'USEPLAN' PARM AS 'USEQUAL'...  
* WITHOUT 'SETPLAN YES', THE FEATURE IS NOT OPERATIONAL, AND THE_  
* INSERTION HAS NO EFFECT ON PROCESSING -- BOTH PARMS ARE REQUIRED.  
*  
* NOTE: USERS MAY SWITCH THEIR AUTHIDS (HLQ) BY SPECIFYING UNDER QMF:_  
*  
* SET CURRENT AUTHID = 'NEWID'  
*  
* THIS WILL SET A NEW VALUE IN THE 'QUALIFY' AND 'USEPLAN' PARAMETERS.  
* ******************************************************
* 

---

**Figure 16. Default host parameter set for DB2 SQL Performance Analyzer under QMF**

b. **Modify each parameter for your target host system.**

The bottom slots are reserved for the future and must not be changed. Any errors in modifying this parameter list can cause unpredictable or incorrect results. The length of the parameter list must remain at 800 characters. Be sure to note the spacing and spelling of parameters.
Tip: Refer to “ANLPARM user parameter settings for QMF” on page 279 for the required settings for using DB2 SQL Performance Analyzer with QMF. For optimal results, modify the parameters in the exit to correspond to the specific host machine that is running QMF with DB2 SQL Performance Analyzer.

3. Optional: Force an unconditional cancellation of queries that exceed the defined limits. If you do not complete this step, DB2 SQL Performance Analyzer gives you the option of canceling each query that exceeds the defined limits. To force an unconditional cancellation of queries that exceed the defined limits:
   a. Find the following code fragment in the ANLGOV1 program:

   ```
   LOGUSER ST R15,XCBERRRET PUT RETURN CODE IN XCB
   MVC XCBMGXT,XANLERO1 ANLGOV1 LIMITS MESSAGE
   MVI XCBLOGM,XCBLOGMY SET QMF TO LOG MESSAGE
   * (ALT) B EXIT16 FOR ALWAYS CANCEL, TAKE EXIT16 NOW
   ```
   b. Delete the asterisk (*) and the (ALT) from the EXIT16 instruction.

4. Optional: Test the QMF governor exits before putting them into production.
   a. Preallocate ISPLLIB (or the ISPF load library) to the test library that you chose by using the standard TSO mode without ISPF and entering the following command or equivalent:

   ```
   ALLOC F(ISPLLIB) DA(test.library) SHR REUS
   ```
b. Restart ISPF and select the QMF option that starts the ANLDSQ1 CLIST or equivalent. When you invoke a QMF query, the DB2 SQL Performance Analyzer Governor Interface program intercepts the query and provides the cost analysis directly to the panel.

c. Make any further changes to the settings for the governor exits based on your test results.

d. When you have completed the changes, migrate the ANLGOV1 and DSQUEGV1 modules into the production library for QMF, which is typically SYS1.SDSQLOAD.

7. Verify that the ANLCDSQ1 CLIST includes the data sets that are currently used for your QMF and GDDM environments. Depending on which release of QMF you are currently running, the actual library names and high-level qualifiers might differ. In the ANLCDSQ1 CLIST, SYS3 is used as the high-level qualifier for most data sets.

<table>
<thead>
<tr>
<th>File name</th>
<th>Sample data set name</th>
<th>Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPPLIB</td>
<td>SYS3.SDSQPJB</td>
<td>QMF ISPF panel library</td>
</tr>
<tr>
<td>ISPMLIB</td>
<td>SYS3.SDSQMLBE</td>
<td>QMF ISPF message library</td>
</tr>
<tr>
<td>ISPSLIB</td>
<td>SYS3.SDSQSLBE</td>
<td>QMF ISPF skeleton library</td>
</tr>
<tr>
<td>SYSEXEC</td>
<td>SYS3.SDSQEXCE</td>
<td>TSO REXX execution library</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>* (terminal)</td>
<td>ANLGOV1 and ANLQMF messages</td>
</tr>
<tr>
<td>DSQPRINT</td>
<td>SYSOUT(X)</td>
<td>Print data output to hold class (X)</td>
</tr>
<tr>
<td>DSQDEBUG</td>
<td>SYSOUT(X)</td>
<td>Trace dump output to hold class (X)</td>
</tr>
<tr>
<td>DSQEDIT</td>
<td>work space</td>
<td>Edit transfer file (temp)</td>
</tr>
<tr>
<td>DSQUDUMP</td>
<td>SYSOUT(X)</td>
<td>Output snap dump to hold class (X)</td>
</tr>
<tr>
<td>DSQSPILL</td>
<td>work space</td>
<td>Overflow spill file (temp)</td>
</tr>
<tr>
<td>DSQUCFRM</td>
<td>DSQUCFRM</td>
<td>Saves user-defined ICUFORMS</td>
</tr>
<tr>
<td>DSNPNLE</td>
<td>SYS3.DSQPNL</td>
<td>QMF panel file</td>
</tr>
<tr>
<td>ADMGMAP</td>
<td>SYS3.DSQMAGE</td>
<td>GDDM map group for QMF</td>
</tr>
<tr>
<td>ADMCFORM</td>
<td>SYS3.DSQCCHART</td>
<td>GDDM chart forms for QMF</td>
</tr>
<tr>
<td>ADMCDATA</td>
<td>SYS3.ADMCDATA</td>
<td>GDDM data file</td>
</tr>
<tr>
<td>ADMGDF</td>
<td>SYS3.ADMGDF</td>
<td>GDDM form definitions file</td>
</tr>
<tr>
<td>ADMSYMBS</td>
<td>SYS3.ADMSYMBS</td>
<td>GDDM symbols file</td>
</tr>
<tr>
<td>ADMPRTNTQ</td>
<td>SYS3.ADMPRTNT.QUEUE</td>
<td>GDDM output print file</td>
</tr>
<tr>
<td>ADMDEF</td>
<td>SYS3.GDDMDEF1</td>
<td>GDDM definitions file</td>
</tr>
</tbody>
</table>

8. Optional: Set up DB2 SQL Performance Analyzer so that you can start it from an existing QMF interface, rather than starting QMF through DB2 SQL Performance Analyzer.

a. Add the following allocations to your startup CLIST (or REXX exec):

```plaintext
ALLOC FI(DSQLLIB) DA('QMF###.SDSQLOAD', 'sysx.hiqual.SANLLOAD', + 'DS###.SDSNEXIT', 'DSN###.SDSNLOAD') SHR REUSE
ISPEEXEC LIBDEF ISPLLIB LIBRARY ID(DSQLLIB)
ALLOC FI(SYSPRINT) DA(*) SHR
```
These statements cause the QMF load library, DB2 SQL Performance Analyzer program load library, and DB2 exit and load libraries to become available.

b. If you install the modified QMF Governor exit into a different library, add the library to the list of allocated libraries as the first entry. The SYSPRINT allocation is required by DB2 SQL Performance Analyzer to direct warning messages to the terminal.

c. Optionally, use the following code at the end of your QMF CLIST (or REXX exec) to free the allocated libraries.

```sql
ISPEXEC LIBDEF ISPLLIB LIBRARY ID( )
FREE FI(DSQLLIB)
```

---

**Optional: Interfacing with DB2 Administration Tool**

If DB2 Administration Tool (DB2 Admin) is installed, you can integrate it with DB2 SQL Performance Analyzer by using Tools Customizer to run the Add SQL PA to DB2 Admin Launchpad task.

**About this task**

A REXX exec, ANLCADBI, is provided in the `high_level_qualifier.SANLCLST` library which allows DB2 SQL Performance Analyzer to interface with DB2 Admin.

**Procedure**

1. Start Tools Customizer.
2. Navigate to the Product Parameters panel from the Customizer Workplace panel.
3. Specify values for all of the parameters listed under the Add SQL PA to DB2 Admin Launchpad task.
4. Generate the customization job.
5. Optional. If DB2 SQL Performance Analyzer will be the first entry to the DB2 Admin, run DB2 Admin's ADBL with the parameter `DMT`.

```sql
EXEC hlq.SADBCLST('ADBL') 'DMT'
```

A dialog box displays. Press enter to confirm the initial table values and PF3 to exit.

This step builds the required Data Management Tools table, and ANLCADBI adds the DB2 SQL Performance Analyzer entry to the DB2 Admin Launchpad.
6. Run the customization job.

**Example**

Running the DB2 Admin CLIST ADBL displays the Data Management Tools Launchpad panel, as shown in the following figure.
Optional: Set up ANLCSPA for use from Browse or Edit sessions

To invoke SQL PA from a browse or edit session, you can use Tools Customizer to run the "Set up ANLCSPA for use from Browse or Edit sessions" task, or you can manually run a CLIST to allocate the CLIST library to SYSPROC.

About this task

If you want to use Tools Customizer, the Set up ANLCSPA for use from Browse or Edit sessions task contains the following optional steps, which you can implement in any combination that you want.

- Create a library that is variable length and it copies all the members in SANLCLST into the new library, which is named SANLVCLS. You can use this library in an allocation to SYSPROC.
- Copy ANLCSPA to a library that is already allocated to SYSPROC. If you do not want to allocate the entire library, you can copy ANLCSPA to a different library.

To invoke ANLCSPA from any browse or edit session, the CLIST library must be allocated to SYSPROC. The preferred method to invoke ANLCSPA is through the TSO logon procedure. The CLIST ANLCALLC is provided for you to use, if the allocation is necessary after the logon procedure.

Procedure

1. Optional: Use Tools Customizer to generate one or more jobs that will help to set up ANLCSPA.
   a. Start Tools Customizer.
   b. Navigate to the Product Parameters panel from the Customizer Workplace panel.
   c. Select the Set up ANLCSPA for use from Browse or Edit sessions task.
   d. Select one or more of the optional steps.
   e. Specify values for all of the required parameters.
   f. Generate the customization jobs.
   g. Run the customization jobs.
2. Optional: Manually run ANLCALLC in the SANLCLST library to allocate the SANLCLST library to SYSPROC.

**Optional: Enabling the stored procedure customization for DB2 SQL Performance Analyzer**

You must complete several tasks if you used the optional Tools Customizer task to enable the use of stored procedures for DB2 SQL Performance Analyzer.

**Important requirements for using the stored procedure**

Using the stored procedure in DB2 SQL Performance Analyzer has some requirements.

A requirement for using the DB2 SQL Performance Analyzer stored procedure is that programs that call any stored procedure must be compiled and link edited with Language Environment (LE) compatible libraries. If LE is not used, the programs fail to initialize properly. See the Language Environment Programming Guide for more information.

To maintain good program performance, the maximum size of an SQL statement supported by the SQL PA stored procedure is 32,000 bytes. For larger SQL statements, consider using the TSO or batch program interfaces for evaluation.

**Requirements for allocating the stored procedure load library**

You can change the load library that is allocated for the stored procedure.

WLM stored procedure address spaces are started by MVS™ automatically, but they also have STEPLIB allocations. Stored procedures must run from an allocated STEPLIB, which must not be the `high_level_qualifier.SANLLOAD` library. The DB2 SQL Performance Analyzer stored procedure load modules are copied to the `high_level_qualifier.SANLLODS` library during customization.

**Restrictions:**

- Do not place ANLPRCR in a library that is on the MVS link list.
- This library must not be APF-authorized.
- Stored procedures must run from an allocated STEPLIB, which must not be the `high_level_qualifier.SANLLOAD` library.

**Customizing the stored procedure address space startup procedure**

You can customize the DB2 stored procedure address space to run the DB2 SQL Performance Analyzer stored procedure.

**About this task**

DB2 SQL Performance Analyzer provides a sample startup procedure named `ssidWLMR` in the `high_level_qualifier.SANLJCL` library.

The stored procedure address space startup procedure JCL must contain an ANLCNTL DD statement that refers to the set of standard ANLCNTL parameters that are normally used by DB2 SQL Performance Analyzer. Also, the address space startup procedure must contain DD cards for SDSNLOAD, SDSNEXIT, and SCEERUN. For a stored procedure to read a non-DB2 file, the file must be
allocated in the stored procedure address space startup JCL. The stored procedures ANLPRER and ANLPRCR can write messages to SYSPRINT. SYSPRINT must be allocated to the stored procedure address space to receive those messages. The load library that contains the DB2 SQL Performance Analyzer stored procedure must be part of the STEPLIBs allocated and be separate from the TSO/batch load library.

To customize this startup procedure (if you do not use ssidWLMR):

**Procedure**

- If ANLPRER or ANLPRCR was loaded into a library that is not already defined by the STEPLIB DD statement, concatenate the library that contains the ANLPRER or ANLPRCR DB2 SQL Performance Analyzer stored procedure to the STEPLIB DD statement.

**Restrictions:**

- Do not place ANLPRCR or ANLPRER in a library that is on the MVS link list.
- This library must not be APF-authorized.
- Stored procedures must run from an allocated STEPLIB, which must not be the high_level_qualifier.SANLLOAD library.

- Add a //SYSPRINT DD SYSOUT=* statement to the address space startup procedure to direct any messages written by the DB2 SQL Performance Analyzer stored procedure.

- Add a //ANLCNTL DD DSN=file_name,DISP=SHR statement to the address space startup procedure to allocate a file that contains a set of configuration parameters for use by all DB2 SQL Performance Analyzer stored procedure users. The data set cannot be the same ANLCNTL data set that is used elsewhere for DB2 SQL Performance Analyzer batch and TSO operations. After the address space is active, the data set (file_name) is considered in use and cannot be modified. Edit the ssidCNTL member, which is located in high_level_qualifier.SANLDATA, and customize those parameters to describe the host system for the stored procedure.

- Insert rows into the catalog.

**Results**

After you have customized the startup procedure, customize the stored procedure ANLCNTL parameters.

**Customizing the stored procedure ANLCNTL parameters**

After you have created, customized, and inserted rows into the catalog for the DB2 SQL Performance Analyzer stored procedures, customize the parameters for ANLCNTL.

**Procedure**

Examine the parameters in ssidCNTL in the high_level_qualifier.SANLDATA library that you used when you customized the stored procedure address space startup procedure, and customize them to describe the target host configuration that you want, set the limits, warnings, and cost algorithm. For more information about the ANLCNTL configuration parameters, refer to “ANLCNTL configuration parameters” on page 262.
What to do next

Start the stored procedure address space.

Calling the DB2 SQL Performance Analyzer stored procedure

After the stored procedure address space has been started, you can call the DB2 SQL Performance Analyzer stored procedure.

About this task

A call to the DB2 SQL Performance Analyzer stored procedure for cost analysis of any SQL statement is accomplished by passing a parameter string. The parameter string for ANLPRCR consists of the names of 11 host variables. The host variables are positional, which means that the stored procedure expects a specific data type and value for a variable depending on its position in the string. The host variable names that are used are arbitrary.

Restriction: All programs that call the DB2 SQL Performance Analyzer stored procedure must be compiled and link edited with LE/370 compatible libraries. If not, the program fails to initialize properly.

The following example shows a typical call to the ANLPRCR stored procedure.

```sql
EXEC SQL CALL ANLPRCR ( :SQL_STMT, :SQL_LEN, :ANL_PARM,
:ANL_CPU, :ANL_ELAP, :ANL_IOC, :ANL_QUN, :ANL_MONY, :ANL_WARN,
:ANL_CODE, :SQL_CODE );
```

The following table uses the names in the example to describe the characteristics and use of each host variable.

<table>
<thead>
<tr>
<th>Host variable</th>
<th>Input or output</th>
<th>Data type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 SQL_STMT</td>
<td>Input</td>
<td>VARCHAR(32000)</td>
<td>Contains the SQL SELECT, INSERT, UPDATE, or DELETE statement to be evaluated, up to 32000 characters long.</td>
</tr>
<tr>
<td>2 SQL_LEN</td>
<td>Input</td>
<td>SMALLINT</td>
<td>Contains the length of the SQL statement.</td>
</tr>
<tr>
<td>3 ANL_PARM</td>
<td>Input</td>
<td>CHAR(240)</td>
<td>Contains up to 15 DB2 SQL Performance Analyzer user parameters. See &quot;Specifying user parameters to the DB2 SQL Performance Analyzer stored procedure&quot; on page 103.</td>
</tr>
<tr>
<td>4 ANL_CPU</td>
<td>Output</td>
<td>DOUBLE</td>
<td>ANLPRCR returns the CPU time estimate in seconds in floating point format (8 byte, double precision).</td>
</tr>
<tr>
<td>5 ANL_ELAP</td>
<td>Output</td>
<td>DOUBLE</td>
<td>ANLPRCR returns the elapsed time estimate in seconds in floating point format.</td>
</tr>
<tr>
<td>6 ANL_IOC</td>
<td>Output</td>
<td>DOUBLE</td>
<td>ANLPRCR returns the I/O count estimate in floating point format.</td>
</tr>
<tr>
<td>7 ANL_QUN</td>
<td>Output</td>
<td>DOUBLE</td>
<td>ANLPRCR returns the DB2 SQL Performance Analyzer QUNITS (query units) estimate in floating point format.</td>
</tr>
</tbody>
</table>
Table 12. SQL PA stored procedure call parameters (continued)

<table>
<thead>
<tr>
<th>Host variable</th>
<th>Input or output</th>
<th>Data type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 ANL_MONY</td>
<td>Output</td>
<td>DOUBLE</td>
<td>ANLPRCR returns the monetary cost estimate in floating point format.</td>
</tr>
<tr>
<td>9 ANL_WARN</td>
<td>Output</td>
<td>CHAR(5)</td>
<td>ANLPRCR returns a string of five (5) Y/N flags for the CPU time, elapsed time, I/O count, QUNITs, and money estimates, in that order. If an estimate exceeds the installation limit, the flag contains a ‘Y’.</td>
</tr>
<tr>
<td>10 ANL_CODE</td>
<td>Output</td>
<td>INTEGER</td>
<td>Contains the return code from ANLPRCR. The code value is 0 if no limits were exceeded, or 16 if one or more limits were exceeded. On certain errors, it might contain another positive integer.</td>
</tr>
<tr>
<td>11 SQL_CODE</td>
<td>Output</td>
<td>INTEGER</td>
<td>Contains the last SQL code from the DB2 SQL Performance Analyzer PREPARE or EXPLAIN statement.</td>
</tr>
</tbody>
</table>

The call parameter list for the EXPLAIN-capable stored procedures (ANLPRER) is slightly different, as shown in the following figure. The fields marked with an asterisk (*) are unique to the ANLPRER stored procedure.

```
EXEC SQL CALL ANLPRER (:SQL_STMT, :SQL_LEN, :ANL_PARM, 
    :ANL_CPU, :ANL_ELAP, :ANL_IOC, 
    :ANL_QUNT, :ANL_COST, :ANL_WARN, 
    :ANL_CODE, :SQL_CODE, :DB2_EXPL, 
    :DB2_RECS);
```

Figure 18. ANLPRER call parameter list

**Specifying user parameters to the DB2 SQL Performance Analyzer stored procedure**

The DB2 SQL Performance Analyzer stored procedure call parameter list contains a list of DB2 SQL Performance Analyzer user parameters in the third host variable. You can specify up to 15 user parameters in this list.
Procedure

1. To specify user parameters to the DB2 SQL Performance Analyzer stored procedure rather than using the default settings, perform the following steps:
   a. Enter the name and the value for the SQL PA parameter, preceded and followed by at least one blank.
      The following set of user parameters are supplied by default:
      
      | Parameter  | Value       |
      |------------|-------------|
      | REPORTS    | STP (STP is required) |
      | VERSION    | V10R1       |
      | STORAGE    | 3390-3      |
      | REFRESH    | ANY         |
      | DEGREES    | 1           |
      | QUALIFY    | authid      |
      | CONNECT    | CAF (or RRSAF) |
      | PRECISE    | YES         |

      Requirement: The first parameter must be the REPORTS STP parameter.
      The list is 240 bytes long. You can supply up to 15 parameter and value sets. Each can be a maximum of 16 characters long, including blanks.
      The default value for the VERSION parameter, which indicates the version of DB2 that you are using, is V10R1. If you use a different version of DB2, you must edit the VERSION parameter, especially if no PLAN_TABLE exists. Otherwise, the parameters are automatically created in DB2 10 format.

2. To report access plans and catalog statistics in the ANLPRER program, add the RETPLAN YES parameter to the passed parameter list.

Example

The following figure shows a sample of the formatted output.
* CAF OPEN RETCODE IS 0
* EXPLAIN PLAN FOR (LENGTH 480)  
* SELECT COUNT(*) FROM  
* SYSIBM.SYSPACKSTM A, SYSIBM.SYSPACKDEP B  
* WHERE A.LOCATION = ?A AND  
* B.DLOCATION = ?A AND  
* B.DCOLLID = ?A AND  
* A.COLLID <> ?A
* ANLPRER SQLCODE IS 0
* ANLPRER RETURNS => WARNING FLAGS: --Y-
ELAPSED: 10.34282 CPU TIME: 7.24000
I/O COUNT: 6 QUNITS: 160886
MONETARY: 1.61
ANL CODE: 0 SQL CODE: 0
RECORDS: 2

The DB2 Access Plan:
Qry: 100000001 Blk: 1 Pln: 1 Mth: 0 Typ: Mix: 0 Rid: Bif:
Acc: 1 Tno: 1 Table: SYSIBM . SYSPACKSTM Corr: A
Index: SYSIBM . DSNKDX01 Ixo: Y Mtch: 2 SortC|N UJOG: NNNN NNNN
Jon: Mrg: -1 Adq: -1 Aid: -1 Jdg: -1 Jid: -1 Cid: -1 Nid: -1 Par:
Rng: Typ: T Enc: CCSID S|M|D: 0 0 0 Coll: SQLPA
Grp: Pref: S Lock: IS App: Pgm: ANLPRER Ver:

The DB2 Statistics used:
Tty: T Loc: Ncol: 18 Rowz: 404 CARD: 212216
Npag: 14830 Pctp: 90 COMP: 0 Encd: E Tsty: Part: 0
Nidx: 1 Pgsz: 4 Ntab: 8 Nact: 15543 Sgsz: 4 Tclo: N
Lkpt: N Maxr: 255 Lksz: A Crat: 0.000000 Cleq: N Cing: Y Ixsz: 4096
Iclo: N Leaf: -1 Ikey: -1 Fkey: -1
Nlvl: -1 Kcol: 5 Iunq: D Ityp: 2 Piec: 0 Log: Y
Tver: 0 Iver: 0 Nmqt: 0 Ailn: 0 Arln: 404
Dsziz: 0 Pools: BP0 BP0 Volt: N Ipad: - Dpsi: 0 Irel: Y

Proc MS: 7240 Frcq SC: 160886 Category: A Reason: NORMAL

The DB2 Access Plan:
Qry: 100000001 Blk: 1 Pln: 2 Mth: 1 Typ: Mix: 0 Rid: Bif:
Acc: 1 Tno: 2 Table: SYSIBM . SYSPACKDEP Corr: B
Index: SYSIBM . DSNKDX01 Ixo: Y Mtch: 2 SortC|N UJOG: NNNN NNNN
Jon: Mrg: -1 Adq: -1 Aid: -1 Jdg: -1 Jid: -1 Cid: -1 Nid: -1 Par:
Rng: Typ: T Enc: CCSID S|M|D: 0 0 0 Coll: SQLPA
Grp: Pref: S Lock: IS App: Pgm: ANLPRER Ver:

The DB2 Statistics used:
Tty: T Loc: Ncol: 10 Rowz: 97 CARD: 11829
Npag: 292 Pctp: 90 COMP: 0 Encd: E Tsty: Part: 0
Nidx: 3 Pgsz: 4 Ntab: 8 Nact: 15543 Sgsz: 4 Tclo: N
Lkpt: N Maxr: 255 Lksz: A Crat: 0.000000 Cleq: N Cing: Y Ixsz: 4096
Iclo: N Leaf: -1 Ikey: -1 Fkey: -1
Tver: 0 Iver: 0 Nmqt: 0 Ailn: 0 Arln: 97
Dsziz: 0 Pools: BP0 BP0 Volt: N Ipad: - Dpsi: 0 Irel: Y

Proc MS: 7240 Proc SU: 160886 Category: A Reason: NORMAL

Figure 19. RETPLAN YES sample output
Migrating DB2 SQL Performance Analyzer from one DB2 version to another DB2 version

After you have migrated your DB2 subsystem, you must configure DB2 SQL Performance Analyzer for the migrated DB2 subsystem.

Related concepts:
- Chapter 2, “Customizing DB2 SQL Performance Analyzer,” on page 23

After you install DB2 SQL Performance Analyzer by following the installation instructions in the Program Directory, you must run Tools Customizer to specify the variables for each DB2 subsystem and to customize the configuration and user parameters.

Migrating DB2 SQL Performance Analyzer to DB2 Version 9.1 for z/OS

To migrate DB2 SQL Performance Analyzer to DB2 Version 9.1 for z/OS, complete the following steps.

Procedure
1. Use Tools Customizer to edit the DB2 parameters for the migrated subsystems.
2. Regenerate the customization jobs.
3. Submit only the jobs that are customized from the ANLDROPD template, the ANLICRE8 template, and the ANLBNDGR template as identified in the TEMPLATE column on the Finish Product Customization panel.

Important: Submitting the job to drop the DB2 objects will delete all EXPLAIN tables in the DB2 SQL Performance Analyzer database, and you will lose DB2 SQL Performance Analyzer history. If you need the existing data in the EXPLAIN tables, save the data before you submit the job, and restore the data after the objects are created.

Migrating DB2 SQL Performance Analyzer to DB2 10

To migrate DB2 SQL Performance Analyzer to DB2 10, complete the following steps.

About this task

Attention: Enabling-new-function mode (ENFM) is not supported.

In DB2 10, EXPLAIN tables that are in pre-Version 8 format are no longer supported. EXPLAIN tables that are in Version 8 or Version 9 format and EBCDIC-encoded EXPLAIN tables are deprecated.

For more information about the restrictions for EXPLAIN tables in DB2 10 and about jobs for completing this task, see the DB2 10 for z/OS Installation and Migration Guide.

Procedure
1. Convert your EXPLAIN tables to UNICODE encoding. If you converted your EXPLAIN tables when you migrated your DB2 subsystems to DB2 10, skip this step.
a. Edit and submit the *high-level-qualifier* SDSNSAMP(DSNTIJXA) job. This job changes the format of the EXPLAIN tables to the format of the current DB2 version.

b. Edit and submit the *high-level-qualifier* SDSNSAMP(DSNTIJXB) job. This job generates SQL to create the new objects in UNICODE, rename the existing EXPLAIN tables, and insert the data from these tables into the new tables.

c. Run the generated SQL from SPUFI or edit the *high-level-qualifier* SDSNSAMP(DSNTIXJC) job. This job uses the DB2 cross-loader control statements that are generated in the DSNTIJXB job.

2. Use Tools Customizer to customize DB2 SQL Performance Analyzer for DB2 10.
   a. Edit the DB2 parameters for the migrated subsystems.
   b. Regenerate the customization jobs.
   c. Submit only the jobs that are customized from the ANLDROPD template, the ANLICRE8 template, and the ANLBNDGR template as identified in the TEMPLATE column on the Finish Product Customization panel.

   **Important:** Submitting the job to drop the DB2 objects will delete all EXPLAIN tables in the DB2 SQL Performance Analyzer database, and you will lose DB2 SQL Performance Analyzer history. If you need the existing data in the EXPLAIN tables, save the data before you submit the job, and restore the data after the objects are created.
Part 3. Administration

The topics in this section provide you with information on administering the product.
Chapter 3. Running DB2 SQL Performance Analyzer

You can run SQL PA as a batch job or interactively under TSO using ISPF panels.

In either case, processing is similar and the reporting options are the same. SQL PA also runs under QMF as an interceptor, providing costing information for SQL statements in flight, prior to running. It can be invoked from any program with DB2 access by calling the SQL PA stored procedure or from any browse or edit session by using the ANLCSPA macro.

DB2 SQL Performance Analyzer can analyze any DB2 runtime environment—users run DB2 SQL Performance Analyzer with a specific configuration (the target host system) and application scenario. These different environments are described by the catalog on the system where processing takes place. Catalog statistics are set to represent production volumes for a full production system while it is still under development. SQL PA users can forecast performance of SQL under CICS, IMS, batch or any other attach facility, on any real or virtual target host machine.

Topics:
- “Using DB2 SQL Performance Analyzer in batch”
- “Running DB2 SQL Performance Analyzer using the ISPF interface” on page 114
- “Using DB2 SQL Performance Analyzer in QMF” on page 134
- “Using the DB2 SQL Performance Analyzer stored procedure” on page 136
- “Easy Explain report levels” on page 138
- “EEEPATH table” on page 140
- “SQL data file: user_ID.ANLEEE.SQL” on page 140
- “TOPA parameter” on page 140
- “EXPLAIN parameters” on page 141
- “VIADRDA remote connect” on page 141

Using DB2 SQL Performance Analyzer in batch

In batch, SQL PA is run as a single step. Input parameter files, SQL and work data sets, and reports, are defined in this information.

Topics:
- “SQL PA Batch JCL”
- “Generating JCL for batch processing” on page 113
- “Required ddnames” on page 113
- “Using debug batch JCL” on page 114
- “Sort Query Limits Report batch JCL” on page 114

SQL PA Batch JCL

You can use this sample JCL for batch processing of your SQL PA job.
The sample JCL in the following figure shows the allocations necessary to run DB2 SQL Performance Analyzer under a sample user ID of TDT690. It includes an optional first step that deletes information left after a previous run.

```plaintext
//JOBNAME JOB (ACCTG), 'SQL/PA 410', CLASS=E, MSGCLASS=X, NOTIFY=USERID

//**********************************************************************
/* SQL/PA -- SQL PERFORMANCE ANALYZER */
//**********************************************************************
/* PROGRAM PROPERTY OF IBM CORPORATION PRODUCT NUMBER 5655-W60 */
/* (C) COPYRIGHT 2000-2010 BY IBM (C) 1993-2005 BY IMSI */
/* SQL PA IS AN IBM LICENSED PROGRAM ALL RIGHTS RESERVED WORLDWIDE */
/* */
/* ALL DB2 RELEASES USE PROGRAM 'ANLSQLPA' */
/* */
//**********************************************************************
/* STEP 0: DELETE THE PREVIOUS RUN'S PERMANENT REPORTS, IF NECESSARY */
/* */
/*ANLSTEP0 EXEC PGM=IEFBR14 */
/*GOAWAY1 DD DSN=&SYSUID..ANLREP.RPT,DISP=(MOD,DELETE), */
/*   SPACE=(TRK,(1,1),RLSE) */
/*GOAWAY2 DD DSN=&SYSUID..QTRACE.RPT,DISP=(MOD,DELETE), */
/*   SPACE=(TRK,(1,1),RLSE) */
/*GOAWAY3 DD DSN=&SYSUID..QLIMIT.RPT,DISP=(MOD,DELETE), */
/*   SPACE=(TRK,(1,1),RLSE) */
/*GOAWAY4 DD DSN=&SYSUID..ANLOUT.RPT,DISP=(MOD,DELETE), */
/*   SPACE=(TRK,(1,1),RLSE) */
/* */
//**********************************************************************
/*ANLSTEP1 EXEC PGM=ANLSQLPA,REGION=0M */
/*STEPLIB DD DSN=SYSX.ANL420.SANLLOAD,DISP=SHR SQL PA */
/*   DSN=SYS1.CEE.SCEERUN,DISP=SHR LE */
/*   DSN=SYS1.DSN###.SDSNLOAD,DISP=SHR DB2 */
/* */
/*SNTRACE DD SYSOUT=* TRACE */
/*ANLCNTL DD DSN=SYSX.ANL420.SANLPARM(SSID CNTL),DISP=SHR */
/*ANLPARM DD DSN=SYSX.ANL420.SANLPARM(SSID PARMA),DISP=SHR */
/****** */
/*ANLWORK DD DSN=&ANLWORK,DISP=(,DELETE,DELETE), */
/*   SPACE=(CYL,(5,1),RLSE),DCB=(LRECL=80,RECFM=FB,BLKSIZE=4000) */
/****** */
/*ANLIN DD DSN=SYSX.ANL420.SANLSQL(ANLSTEST),DISP=SHR */
/****** */
/*ANLOUT DD DSN=&SYSUID..ANLOUT.RPT,DISP=(,CATLG,DELETE), */
/*   SPACE=(CYL,(1,1),RLSE),DCB=(LRECL=80,RECFM=FB,BLKSIZE=9040) */
/* */
/*SYSPRINT DD SYSOUT=*,DCB=LRECL=133 */
/*ANLPRINT DD SYSOUT=*,DCB=LRECL=133 */
/* */
//**********************************************************************
/* ANLREP IS THE EXPLAIN REPORT: YOU MAY WISH TO NAME IT SO THAT */
/* YOU CAN RECOGNIZE WHICH INPUT SQL PRODUCED THIS PARTICULAR REPORT. */
/* */
/* QTRACE IS THE DETAILED TRACE REPORT: WRITTEN WHEN "ALL" REPORTS */
/* ARE REQUESTED. CONTAINS A MORE DETAILED EXECUTION FORECAST. */
/* */
/* QLIMIT IS THE QUERY LIMITS REPORT: IT CONTAINS A FLAG FOR EACH */
/* LIMIT EXCEEDED, PLUS COST VALUES, ONE LINE PER QUERYNO. */
/* */
//**********************************************************************
/*ANLREP DD DSN=&SYSUID..ANLREP.RPT,DISP=(,CATLG,DELETE), */
/*   SPACE=(CYL,(1,1),RLSE),DCB=(LRECL=132,RECFM=FB,BLKSIZE=23760) */
/*QTRACE DD DSN=&SYSUID..QTRACE.RPT,DISP=(,CATLG,DELETE), */```
Generating JCL for batch processing

You can use the BATCH command to generate JCL to be used in batch processing.

Procedure

1. From any processing panel off the basic processing panel (these panels include ANLPPLAN, ANLPPKGE, ANLPDBRM, ANLPFILE, ANLPQNO, ANLPQMF, and ANLPTBLS), enter the BATCH command.

2. Enter the data set to which you want to save the JCL. The JCL for running that job is saved in the data set that you specified.

What to do next

Submit the saved JCL to run the job in batch.

Required ddnames

The following list contains descriptions of the roles of the DD statements.

ANLCNTL

Points to a parameter set. These parameters describe the target host configuration, the costs of various resources in that configuration, and the threshold values for excessive use warning message.

ANLIN

Points to the input data set, which is a sequential file, a member of a partitioned data set (PDS), or a DBRM library member. It can be a file containing SQL statements created by Path Checker. It can also point to an entire DBRM library (no member name is specified) if the DBRMKEY parameter is used to identify the member or members to be processed.

The input source is scanned by ANLSQLPA and all executable SQL statements are extracted from that file's contents. These SQL statements are then submitted to DB2 for the latest Explain, using the current catalog statistics. The tables and indexes chosen on the access path are referenced in the catalog to retrieve the necessary sizing information.

ANLOUT

Contains the actual SQL processed by SQL PA along with a date and time stamp on first record.

ANLPARM

Points to a parameter set. These parameters describe user-specific controls for the application being analyzed, such as the buffer hit ratio, the type attach facility, and the level of reporting by SQL PA.

ANLPRINT

Provides the output data set containing the Cost Summary report.

ANLREP

Provides the output data set for Enhanced Explain report. It is required if the ANLPARM parameter REPORTS is set to REP or DET.

ANLWORK

A temporary work file used when multiple DBRM support is activated by
using the DBRMKEY parameter. You can leave the ANLWORK allocation in your batch JCL even if multiple DBRM support is not used.

The DCB parameters for LRECL and RECFM for this file must be set as required by SQL PA.

QLIMIT
Provides the output data set for the QLIMITS (query limits) report.

QTRACE
Provides the output data set for the Detail Trace report. It is required if the ANLPARM parameter REPORTS is set to DET.

STEPLIB
Might be required if the SQL PA modules are not listed in the system link list (LNKLSTxx) when installed. Also, if the load library for the DB2 subsystem is not on the operating link list for the system, it must be added to STEPLIB for the ANLSQLPA job step. The LE runtime library (SCEERUN) can also require a STEPLIB allocation, as will the DB2 load library, particularly if there are multiple DB2 subsystems present.

SYSPRINT
Provides the output data set containing informational and tracing information.

The data set names for the report files are any names you choose. The DCB parameters for LRECL and RECFM must be formatted as required by SQL PA.

Using debug batch JCL
If you encounter problems with a SQL PA operation, you can document the details to send to IBM support. The ssidJBUG job is like the regular batch JCL but contains additional system output to help diagnose problems.

Procedure
1. Reissue SQL PA using the special ssidJBUG batch JCL found in the hiqual.SANLJCL library.
2. Insert the special diagnostic parameter DBTRACE DMP as the first parameter in the ANLCNTL file, when using debug to document a problem. This control card does not produce a dump, but rather dumps the content of program storage to help the diagnosis. It also copies all the output report files from a single batch run (such as ANLREP) to SYSOUT.
3. Collect and send the problem data from the SYSOUT directory.

Sort Query Limits Report batch JCL
To sort the Query Limits Report, run SQL PA by using the ssidJQST batch JCL that is located in the hiqual.SANLJCL library.

About this task
This job is like the regular batch JCL but contains an extra step for sorting.

Running DB2 SQL Performance Analyzer using the ISPF interface
DB2 SQL Performance Analyzer can run interactively under TSO using standard ISPF panels.
This option allows you to work through several scenarios online, saving and reviewing previous results, while evaluating a number of alternatives and their costs. SQL PA provides the same level of information online under TSO that is generated by a batch run. When SQL PA is run online, you can reset key parameters on the panel and re-evaluate SQL dynamically. You can also choose from among the target host systems defined by your installation, applying the SQL PA costing against various configurations which might be encountered in production.

Topics:
- “Starting SQL PA using ISPF”
- “Navigating the DB2 SQL Performance Analyzer panels”
- “Using scrollable fields on DB2 SQL Performance Analyzer panels” on page 116
- “DB2 SQL Performance Analyzer primary menu panel” on page 117
- “Defining output reports” on page 117
- “Modifying DB2 SQL Performance Analyzer parameters” on page 118
- “Basic processing options” on page 119
- “Comparing old and new plans or packages” on page 120
- “ANLCSPA edit macro” on page 121
- “What if? analysis” on page 124
- “DB2 SQL Performance Analyzer statistics migration” on page 130
- “DB2 SQL Performance Analyzer reports menu” on page 133

Starting SQL PA using ISPF
SQL PA using ISPF is invoked by issuing the ANLCANLI command either from standalone TSO, or from ISPF panel 6, the TSO Command Panel.

About this task

For an example of the main selection panel, see Figure 22 on page 117. You can add DB2 SQL Performance Analyzer directly to the existing menu of DB2 options, and invoke it as an option from that panel.

Figure 20 shows the invocation using the ANLCANLI CLIST.

Figure 20. Invoking ANLCANLI

Navigating the DB2 SQL Performance Analyzer panels
All access to DB2 SQL Performance Analyzer online begins at the DB2 SQL Performance Analyzer main control panel.

From that panel, you can process SQL statements from various sources, evaluate the effect of changing DB2 definitions, and modify the DB2 SQL Performance Analyzer configuration. After DB2 SQL Performance Analyzer is invoked, you go to the SQL PA Query Limits Report to review the results.
After entry into the DB2 SQL Performance Analyzer system under TSO the welcome panel is displayed, as shown in Figure 21. Press Enter to display the DB2 SQL Performance Analyzer main control panel.

Using scrollable fields on DB2 SQL Performance Analyzer panels

DB2 SQL Performance Analyzer uses ISPF scrollable fields to accommodate field text that is larger than the viewable area.

About this task

The less than (<) and the greater than (>) symbols denote a scrollable field. A > symbol indicates that the field can be scrolled to the right. A < symbol indicates that the field can be scrolled to the left. Both symbols are displayed when you are in the middle of data and can scroll either left or right.

Procedure

1. To scroll through the field, type LEFT or RIGHT in the command field, position the cursor in the field, and press Enter.
2. To see the entire contents of the field at once, type EXPAND in the command field, position your cursor in the scrollable field, and press Enter.
3. To clear the contents of the field, type ZCLRSFLD in the command field, position your cursor in the scrollable field, and press Enter. If your level of z/OS does not support the ZCLRSFLD command, you can use the EXPAND command to display the entire contents of the field, and then clear the contents of the field in the window.
4. Optional: Assign your PF keys to be the LEFT, RIGHT, EXPAND, and ZCLRSFLD commands. Using a PF key simulates both typing in the command and pressing Enter.
Example

SQLPA420 --------------- Process QMF statement ------------------------- 14:36
Command ===>

Scroll ===>
CSR

Commands: EXPLAIN SQL TABLES
DB2 system: DSNA
DB2 SQLID : JPUBLIC

Enter the query identification:

*Query name. . . . . AQURY8 >
Query owner . . . . JSITH >
Current degree. . . . 1 (1 or Any)
(An * indicates a required field.)

DB2 SQL Performance Analyzer primary menu panel

You begin DB2 SQL Performance Analyzer processing from the DB2 SQL Performance Analyzer primary menu panel.

The DB2 SQL Performance Analyzer primary menu panel shown in Figure 22
From this panel, you can access panels to perform the following actions:
• Modify and restore statistics
• Perform statistics migration
• Change DB2 system parameters
• Change the current DB2 SQLID
• Change DB2 SQL Performance Analyzer parameters
• Define report data sets
• Enter or update batch job card parameters

Defining output reports

For some output reports, you must specify certain files.
The name of the Cost Summary report file is predefined as `prefix.ANLCOST.LOG`, and is allocated by DB2 SQL Performance Analyzer under the running TSO user ID. The prefix is normally the TSO user ID, but it can be set to a different value.

The Plan Table report is allocated here as `ANL420.PRN`.

For Enhanced Explain reports, you must specify an OUTPUT EXPLAIN FILE. In batch, this file was allocated with the ANLREP DD statement. In the example in Figure 23, the file is named `ANL420.EXP` under the user ID of the TSO user.

The Detail Trace file is declared as `ANL420.DET`.

The DB2 SQL Performance Analyzer Limits file is also allocated here as `ANL420.LIM`.

The `prefix.ANLOUT.SQL` file is now allocated and visible as the extract report that contains SQL and the optional objects that are used (tables and indexes). The prefix is normally the TSO user ID, but it can be set to a different value.

**Modifying DB2 SQL Performance Analyzer parameters**

You can view and modify the current values for all DB2 SQL Performance Analyzer parameters from the DB2 SQL Performance Analyzer Parameter Data Sets panel.

**About this task**

This panel shows the data sets which contain the current values for the Easy Explain Report, the user, and the system options.

**Procedure**

1. Specify option 3 on the main menu. The SQL PA Parameter Data Sets panel is displayed.
2. To modify these values, enter YES in their respective fields to edit the current parameter settings.

**Basic processing options**

You can use the Basic Processing Options panel to choose how you want to input SQL statements from various sources.

**About this task**

Figure 25 shows the options for input SQL statements from the various sources.

**Procedure**

1. To process plans or packages, or to compare old and new plans and packages, select option 1 or 2.
2. To process the SQL in DBRM members, select option 3. From the Process DBRM panel, you can select one or more DBRM by leaving the DBRM name
blank or by typing a wildcard. If you are selecting multiple DBRM members, a
panel displays that allows you to select one or more DBRM members from the
list.
3. To process SQL from a sequential file or PDS member, select option 4.
4. To process SQL from a program source, select option 5.
5. To process a query number from the plan table, select option 6.
6. To process a QMF statement, select option 7.
7. To process tables from the DB2 system catalog, select option 8.

Comparing old and new plans or packages
You can use DB2 SQL Performance Analyzer to compare old and new plans and
packages from the DB2 system catalog.

Procedure
1. Set the EXPLAIN option of the BIND command to YES. Figure 26 shows a
   comparison between the current access path and costs and the previously
   stored plan.
2. Specify option 1 Basic DB2 SQL PA Processing, option 1 Process plans from
   the DB2 system catalog. Or option 1 Basic DB2 SQL PA Processing and 2
   Process packages from the DB2 system catalog.
3. Indicate Y in the *Compare the old plan to the new plan? field.

Viewing the last set of generated reports
You can use the LASTREPORTS command to view the last set of generated reports
without re-running the EXPLAIN command.

Procedure
To view the last set of generated reports, issue the LASTREPORTS command on
any of the following panels:
- ANLPLPLAN
If the last set of reports was generated by running the EXPLAIN command on the
ANLPQNO panel, the plan table report is displayed. If the last set of reports was
generated by running the EXPLAIN command on a panel other than the
ANLPQNO panel, the query limits report is displayed.

ANLCSPA edit macro

In ISPF Edit or Browse mode, DB2 SQL Performance Analyzer allows you to
selectively pick a portion of the file for analysis, choosing one or several
statements.

Selective analysis uses a special edit macro called ANLCSPA. ANLCSPA uses the
letter E in the line number columns to selectively pick a subset of the contents of a
file. You can choose to bracket the desired lines between a pair of EE-EE book
markers or indicate the number of lines. An entry of E12, for example, selects the
next 12 lines. Remember to enter ANLCSPA on the command-line prompt so that
ISPF Edit recognizes this special DB2 SQL Performance Analyzer edit macro.

To invoke ANLCSPA from any edit or browse session, the CLIST library must be
allocated to SYSPROC. The preferred method to invoke ANLCSPA is through the
TSO logon procedure. The CLIST ANLCALLC is provided for you to use if the
allocation is necessary after logon.

Tip: Do not mix the formats of the libraries in SYSPROC. If your SYSPROC
allocation uses libraries with variable block formatting, select the optional
step Create SAMLVCNS on the Product Parameters panel in Tools
Customizer. ANLCALLC will allocate the appropriate library, and, if there is a
mismatch, a message will be issued and the allocation will not be done.

If the DB2 SQL Performance Analyzer CLIST library is allocated as SYSPROC, you
can start DB2 SQL Performance Analyzer from any Edit or Browse session.

If you want to use DB2 SQL Performance Analyzer during a programming session
to evaluate an SQL statement, bracket the statements with EE-EE and issue
ANLCSPA on the command line.

The ANLCSPA edit macro offers the capability to have SQL statements, embedded
in a program or SPUFI source, explained directly from the ISPF/PDF editor. The
following languages are supported:

- Assembler
- C/370™
- COBOL
- Fortran
- PL/I

The edit macro is invoked under ISPF command ANLCSPA. When editing the
source of a program (or SPUFI input), you can specify E as the ISPF/PDF editor
line command and then enter ANLCSPA (or another suitable name of a REXX exec) on the command line and ANLCSPA scans the specified range for any SQL statements.

You can use EE on the first and last line, you can use one single E if the statement is on one single line, or you can use E\text{n} where \( n \) is the number of lines to scan. You can even specify E99999 on the first line of the source and ANLCSPA scans the complete source for any SQL statements and then explains the statements that are explainable. The remaining statements display as comments in the Easy Explain summary report. If you do not specify E, EE, or E\text{n}, all the explainable SQL statements in the module are explained.

You can optionally pass the subsystem ID and the HLQ of the DB2 SQL Performance Analyzer installation to the macro. Passing the subsystem ID and the HLQ is helpful if there are several choices where you do not always want to use the default, for example ANLCSPA DSNA, USERS.SQLPA3.

When the edit macro starts, it optionally asks for the language of the source code you are editing. It then gives you the opportunity to specify a table qualifier and the current SQLID (DB2 object qualifier).

Figure 27 shows an example of editing a PL/I program. During editing, you need to explain the SQL statement that starts in line number 19 and ends in line number 25.

Enter EE in the sequence number field of the lines indicating the start and end lines. Further, the name of the REXX exec is specified in the command field.

```
EDIT JOHNDOE.PLI.SOURCE(E3ONLINE) - 01.01
Command ===> anlcspa
****** *********************************** Top of Data ***********************************
0000100 cl0stst: procedure options(main reentrant) reorder;
0000200
0000300 dcl (number, height) integer;
0000400
0000500 dcl 1 ud,
0000600 3 navn char (8),
0000700 3 indikator char (11);
0000800 dcl
0000900 addr builtin,
001000 high builtin,
001100 translate builtin,
001200
001300 dcl plixopt char (26) var init ('ISA(28k), ISAINC(24k),NR') EXT;
001400
001500 dcl i bin fixed (31,0) init (0) static;
EE
001700 exec sql include sqlca;
001800
001900 exec sql
002000 declare C1 cursor for
002100 select count(*)
002200 from sysibm.systables
002300 where creator = USER
002400 and name in ('TAB1','MYTAB','EEE_TAB2')
002500 and type = 'T';
EE
002700 exec sql
****** *********************************** Bottom of Data ***********************************
```

Figure 27. Edit PL/I program example
The next set of sample figures shows how SQL statements in a given DB2 package are explained while you are using the DB2 Administration Tool. Initially, you specify an owner and collection identification:

Issue the SQ command to display all SQL statements for the package E61PLAN. The results are shown in Figure 29.

You now start the ANLCSPA macro. Because no range is specified in the sequence number column, all explainable SQL statements are explained:
What if? analysis

By using the What if? program you can create scenarios with the optimizer by changing table or index statistics to see if you can influence the access path selection. You can display, modify, save, and restore catalog statistics on both tables and indexes that are most effective in access path selection.

You can define the following criteria for tables:
- Cardinality (number of rows, or CARD)
- Number of pages used by the table (NPAGES)
- Number of active pages (NACTIVE) in the table space
- Percentage of pages used (PCTPAGES)
- Degree of data compression (PCTROWCOMP)

You can define the following criteria for indexes:
- Number of index levels in the tree (NLEVELS)
- Total number of index leaf pages (NLEAF)
- Cardinality of the first column of the index (FIRSTKEYCARD)
- Fully qualified key (FULLKEYCARD)
- Cluster ratio, a 0 - 1 decimal number that indicates the degree to which rows are clustered together on the same or adjacent pages (CLUSTERRATIO)

Each TSO user has a DB2 SQL Performance Analyzer program-maintained partitioned data set (PDS) that is called user_ID.ANL420.SANLSTAT that is built under their TSO user ID. The What If? and catalog migration functions use this PDS to create and manage the statistical processing for backups and restores, temporary parameters, and JCL.

This file can also have a different prefix (like ANLCOST.LOG and ANLOUT.SQL).

Analyzing indexes and modifying statistics

You can analyze indexes and modify statistics to create scenarios with the optimizer to try to influence access path selection. You can determine which catalog statistics on both tables and indexes are most effective in access path selection. You can perform a typical What if? index analysis and statistics modification using this procedure.

Procedure

1. Select option 2, Advanced DB2 SQL PA Processing, on the Figure 22 on page 117 The Advanced Processing Options panel is displayed, as shown in the following figure:
2. Select option 1 to display the Modify Statistics panel, as shown in the following figure.

![Figure 31. Advanced Processing Options panel](image1)

3. To modify the current index statistics, select option 2.

![Figure 32. Modify Statistics panel](image2)
4. Type the index creator, index name, and member name.

Creating and dropping indexes

You can create and drop real and virtual indexes. Creating virtual indexes and replacing a real index with a virtual index helps to calculate the cost of SQL queries.

About this task

For example, to calculate the cost of SQL queries, you can drop a real index virtually, then create and drop many virtual indexes to simulate a different index configuration. The virtual drop of a real index causes the EXPLAIN command to exclude a virtually dropped real index from the cost calculation of SQL queries, but it does not remove the physical index from the DB2 catalog. Creating a virtual index causes the EXPLAIN command to include a virtual index for the cost calculation, but it does not require processor resources to build a virtual index even when the table size of the index target is large. These virtual index features help to design more efficient indexes without costing too much processor resource.

In order to include the virtual index as a part of the cost calculation of the EXPLAIN SQL command, the same DB2 SQL ID should be set and displayed at the upper right corner of the Create Index panel and the SQL PA cost analysis panels. In other words, the table owner of PLAN_TABLE and DSN_VIRTUAL_INDEXES must be the same to include a virtual index in the cost calculation of the EXPLAIN SQL command.

Procedure

1. Select option 2, Advanced DB2 SQL PA Processing, on the [Figure 22 on page 117]. The Advanced Processing Options panel is displayed.
2. Select option 3, Create or drop an index, and press Enter. The Create or Drop Index panel is displayed, as shown in the following figure:
3. To create real indexes, complete the following steps:
   a. On the Create or Drop Index panel, specify the name of the index creator and the name of the index to be created.
   b. Select option 1, Create index, and press Enter. The Create Index panel is displayed, as shown in the following figure:

![Create or Drop Index panel](image)

Figure 34. Create or Drop Index panel

   c. Specify the table creator and the name of the table for which the index will be created.
   d. In the Index column 1 field, specify a name for the column and whether the column will be in ascending or descending order, and press Enter. The Create Index panel is displayed, as shown in the following figure:

![Create Index panel](image)

Figure 35. Create Index panel
e. Specify values for all of the following required fields:

**PRQTY**
Specify the minimum primary space allocation for a DB2-managed data set.

**STOGROUP**
Specify the name of a storage group in which to create the index. The storage group must exist at the current server and the privilege set must include SYSADM authority, SYSCtrl authority, or the USE privilege for the storage group.

**SEQTY**
Specify the minimum secondary space allocation for a DB2-managed data set.

**BUFFERPOOL NAME**
Specify the buffer pool to use for the index. The name of the buffer pool must identify an activated 4 KB, 8 KB, 16 KB, or 32 KB buffer pool, and the privilege set must include SYSADM or SYSCtrl authority or the USE privilege for the buffer pool.

f. Optional: Specify values for the following optional fields:

**UNIQUE**
Specify whether the index to be created will be unique.

**CLUSTER INDEX**
Specify whether the index to be created will be the clustering index for the table.

**PADDED**
Specify whether the index to be created will be padded.

**PARTITIONED**
Specify whether the index to be created will be partitioned.

**CLOSE**
Specify whether the index to be created can be closed when it is not being used.

**WHERE NOT NULL**
Specify whether NULL values in a unique index can be unique.
**INCLUDE NULL KEYS**
Specify whether to include NULL key values.

g. Press Enter. The index is created.

4. To create virtual indexes, complete the following steps:
   a. On the Create or Drop Index panel, specify the name of the index creator and the name of the index to be created.
   b. Select option 3, **Create virtual index**, and press Enter. The Create Index panel is displayed.
   c. Specify the table creator and the name of the table for which the virtual index will be created.
   d. In the **Index column 1** field, specify a name for the column and whether the column will be in ascending or descending order, and press Enter. The Create Index panel is displayed, as shown in the following figure:

![Create Index panel](image)

Figure 37. Create Index panel

e. Optional: Specify values for the following optional fields:

**UNIQUE**
Specify whether the index to be created will be unique.

**CLUSTER INDEX**
Specify whether the index to be created will be the clustering index for the table.

**PADDDED**
Specify whether the index to be created will be padded.

**PARTITIONED**
Specify whether the index to be created will be partitioned.

**NLEAF**
Specify the number of active leaf pages in the index.

**NLEVELS**
Specify the number of levels in the index tree.

**PGSIZE**
Specify the size in kilobytes of the leaf pages in the index.

**CLUSTERRATIOF**
Specify the percentage of rows that are in clustering order.
FIRSTKEYCARDF
Specify the number of distinct values of the first key column.

FULLKEYCARDF
Specify the number of distinct values of the key.

f. Press Enter. The virtual index is created.

5. To drop real indexes, complete the following steps:
   a. On the Create or Drop Index panel, specify the name of the index creator and the name of the index to be dropped.
   b. Select option 2, Drop index, and press Enter. A confirmation panel is displayed.
   c. Confirm that you want to drop the index, or cancel the operation.

6. To drop virtual indexes, complete the following steps:
   a. On the Create or Drop Index panel, specify the name of the index creator and the name of the virtual index to be dropped.
   b. Select option 4, Drop index virtually, and press Enter. The specified index is dropped.

Removing virtual changes

Procedure
1. On the Create or Drop Index panel, specify the name of the index creator and the name of the index to be changed.
2. Select option 5 Remove virtual changes, and press Enter. The virtual changes are removed.

Restoring statistics for tables and indexes
You can restore previously saved original statistical sets from a table or an index.

Procedure
1. Select option 2, Advanced DB2 SQL PA Processing, from the [Figure 22 on page 117]. The Advanced Processing Options panel is displayed. [Figure 31 on page 125]
2. Select option 1 to display the Modify Statistics panel, as shown in [Figure 32 on page 125]
3. To restore saved statistics for an index or table, select option 3.

DB2 SQL Performance Analyzer statistics migration
By using the statistics migration function, you can manage the process of running the ANLCAT batch program to migrate statistics and use the ssidAPPL set of JCL in high_level_qualifier.SANLJCL to apply them to the system catalog. The statistics migrations panels dynamically build jobs to Collect or Apply statistics under TSO.

Setting up the Apply and Collect jobs
The Collect and Apply processes use the job card information saved in the user parameters during customization.

About this task
If the (USRParm) Allow users to have a copy of the user parameters parameter was selected on the Product Parameters panel in Tools Customizer, each user can have his or her own job card. Job cards are saved in the user parameter file. If the user-level user parameters have not been set up before the job cards are updated,
the Parameter Data Sets panel will be displayed so that the user-level user parameters can be initialized.

**Procedure**

1. Select option 3, **Modify DB2 SQL PA Configuration**, from the primary menu panel.

2. Select option 5, **Enter or update the batch job card parameters**, to display the Batch Job Statement Parameters panel, as shown in the following figure.

![Batch Job Statement Parameters Panel](image)

**Collecting catalog statistics**

Use the Collect catalog statistics function to specify the collection options and parameters that you want to migrate or apply to the system catalog.

**About this task**

**Recommendation:** It is recommended that you collect a set of statistics for your baseline system before applying any changes to the catalog. Because the data is saved in a member, you can always restore those statistics to their original settings after you have completed your analysis.

**Procedure**

1. Select option 2, **Advanced DB2 SQL PA Processing**, from the primary menu panel. The Advanced Processing Options panel is displayed.

2. Select option 2 to display the Statistics Migration panel, as shown in the Statistics Migration panel.
3. Select option 1, **Collect statistics**, to collect statistics by building the input parameters and executing the ANLCAT program.

**Applying catalog statistics**

Use the Apply catalog statistics function to apply or migrate the collected statistics to the system catalog.

**Procedure**

To apply catalog statistics that are already collected or that were manually created, use the following steps:

1. Select option 2, **Advanced DB2 SQL PA Processing**, from the primary menu panel. The Advanced Processing Options panel is displayed.
2. Select option 2 to display the Statistics Migration panel.
3. Select option 2, **Apply statistics**, to apply statistics by updating the DB2 catalog with collected statistics.

### DB2 SQL Performance Analyzer reports menu

After SQL costing is complete, DB2 SQL Performance Analyzer displays the Query Limits Report. You can browse the reports created during this run from this panel.

You can sort the Query Limits Report by using the **SORT** command. The optional parameters are the sort column number and the sort order. For example, to sort the **Query No** column in ascending order, the command is **SORT 3 A**.

The Query Limits Report panel is shown in the following figure.

![Figure 41. Query Limits Report panel](image-url)
Using DB2 SQL Performance Analyzer in QMF

An optional exit in the QMF Governor allows SQL PA to cost your QMF queries. Using this installation option makes it easy for you to use DB2 SQL Performance Analyzer while developing SQL in QMF.

The QMF interface runs in the background. The QMF interface only displays if the SQL statement that you are about to run is likely to exceed the cost limits defined for DB2 SQL Performance Analyzer. For queries that exceed the cost limits, the QMF interface displays the cost estimate and gives you the chance to stop before QMF begins processing.

Topics:
- “QMF intercept exit”
- “DB2 primary options menu”
- “DB2 SQL Performance Analyzer cost summary under QMF”
- “Run queries that exceed limits in QMF” on page 135
- “QMF batch processing” on page 136

QMF intercept exit

DB2 SQL Performance Analyzer has a special exit used to intercept QMF transactions in flight. The special exit provides you with an immediate cost analysis, before actually running the query. If installed, this option is integrated into your normal QMF processing. To integrate the option, add a call to the SQL PA exit as part of the IBM-supplied QMF Governor’s activities.

DB2 primary options menu

You can invoke QMF from a menu or from a QMF CLIST.

You do not do anything special to use DB2 SQL Performance Analyzer under QMF. The QMF intercept exit allows DB2 SQL Performance Analyzer to take control of a query before it is issued and perform a streamlined cost analysis of the SQL statement.

From the QMF Home panel, users typically create ad hoc queries, composing their SQL in real time.

DB2 SQL Performance Analyzer can provide cost and performance information on each SQL statement before DB2 processes it.

DB2 SQL Performance Analyzer cost summary under QMF

The QMF Intercept Exit for DB2 SQL Performance Analyzer produces the standard Cost Summary Report, along with any warnings that are warranted by the SQL statement.

The three types of costing information displayed for the user to consider are:
- Resource Consumption: CPU time, elapsed time, I/O count
- Relative Cost Valuation: QUNITS (Query Units)
- Financial Valuation: Monetary cost of execution (Charge Back)
The example in the following figure shows this report.

```
+---------------------------------------------------------------------+
| QMF ONLINE QUERY WILL REQUIRE 2.37486 SECONDS OF ELAPSED TIME        |
| DURING WHICH 0.63661 SECONDS OF CPU TIME WILL BE CONSUMED AND       |
| A TOTAL OF 7 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK           |
| QUNITS 9 ESTIMATED PROCESSING COST $ 0.4566 DOLLARS                |
+---------------------------------------------------------------------+

Figure 42. Cost summary under QMF

Run queries that exceed limits in QMF

Based on a customization option, the QMF Intercept Exit can show the cost of every QMF query, or remain invisible, showing the costs only when queries exceed one or more installation limits (CPU time, I/O, elapsed time).

Typically, the choice is to show the costs only when a query exceeds a limit. After presenting the SQL PA Cost Summary report and any warning messages for high usage of resources, the QMF Interceptor Exit asks if you want to cancel the query. If you respond Y (yes), the query is not run. If you respond N (no), processing continues despite the high-cost estimate.

Note: You can configure DB2 SQL Performance Analyzer to prevent queries with excessive cost estimates from running. If a query exceeds a limit, you have no option to run it.

The following figure shows an example of canceling a query with an excessive CPU time estimate.

```
ANL5025W *** WARNING:
ESTIMATE OF 73.30724 EXCEEDS *CPU TIME* LIMIT OF 10 CPU SECONDS!

+---------------------------------------------------------------------+
| QMF ONLINE QUERY WILL REQUIRE 115.90003 SECONDS OF ELAPSED TIME     |
| DURING WHICH 73.30724 SECONDS OF CPU TIME WILL BE CONSUMED AND       |
| A TOTAL OF 784 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK          |
| QUNITS 1012 ESTIMATED PROCESSING COST $ 49.7052 DOLLARS             |
+---------------------------------------------------------------------+

*** WOULD YOU LIKE SQL PA TO CANCEL THIS QUERY (Y/N) ?
Y
***

Figure 43. Canceling a query under QMF

After cancellation, you return to the QMF Query Panel, and a message indicating that the query was canceled by the DB2 SQL Performance Analyzer governor exit displays on the screen, like the following figure.
If the query is not canceled by you or by the install option, QMF proceeds as normal. The following figure shows the Status Panel with the QMF cost estimate. The cost estimate is relative for QMF. It is not the cost estimate for SQL PA.

Finally, the QMF results panel is displayed, showing the result of the query.

**QMF batch processing**

You can use batch processing, if your QMF queries are estimated to have long runtimes and you do not want to run them online.

If the QMF Intercept Exit estimates a long runtime and heavy resource usage for your QMF queries, and you would rather not run them online, you can do one of two things after canceling the query:

1. Structure the query in a different way, perhaps qualifying with additional predicates that might allow indexed access more efficiently.
2. Transform the ad hoc query into a QMF batch job, where it can run in the background while you continue to process other, shorter queries online.

Create batch query jobs, by entering `RUN Q.DSQ1EBIN` on the command line and then following the QMF instructions. See your QMF user manuals for more information about running QMF batch queries.

**Using the DB2 SQL Performance Analyzer stored procedure**

The DB2 SQL Performance Analyzer analysis of SQL statements is available through DB2 stored procedures named ANLPRER or ANLPRCR.

Users from any address space or DB2 Connection can issue a stored procedure call to ANLPRER or ANLPRCR to request the cost of a query before running it. DB2 SQL Performance Analyzer returns the costs of the query, along with warnings and return codes indicating if the query runs within the defined limits.

A DB2 stored procedure is a specialized program, or procedure, that is called from any DB2 application program, including DB2 clients in PCs, and applications in CICS, IMS or other regions. The procedure is called like a subroutine or subprogram, accepting host variables as input, and returning host variables and row sets as output.
The DB2 SQL Performance Analyzer stored procedure ANLPRCR uses the RRSAF method. An entry for ANLPRCR must be placed in the SYSIBM.SYSROUTINES catalog table and describe the characteristics of the procedure, such as its parameter list. The procedure must be bound as a package on the system before applications can call it.

For more information about DB2 stored procedures, refer to the DB2 publications.

Topics:
- Accessing the DB2 SQL Performance Analyzer stored procedure

Accessing the DB2 SQL Performance Analyzer stored procedure

You can access the DB2 SQL Performance Analyzer stored procedures using the supplied PL/I sample programs ANLSTPR and ANLSTER.

The source code for all both programs is located in high_level_qualifier.SANLSAMP. The execution JCL is also provided in high_level_qualifier.SANLJCL.

ANLSTPR test program

You can use the ANLSTPR test program to test your stored procedures.

To start the ANLSTPR test program, run the ssidSTPR JCL job in high_level_qualifier.SANLJCL:

```
//JOBNAME JOB (ACCTG), 'PLI RRSAF', CLASS=1, MSGCLASS=X, NOTIFY=USERID
//*DSN9STPR
//*
//********************************************************************
//** CALLS EXPLAIN-ENHANCED STORED PROCEDURE ANLPRCR (RRSAF)
//** STORED PROCEDURE (ANLPRCR) TEST PROGRAM - ANLSTPR (PL/I) SAMPLE
//**
//** LICENSED MATERIALS - PROPERTY OF IBM CORPORATION - 5655-W60
//** COPYRIGHT IBM CORPORATION 2000, 2013 ALL RIGHTS RESERVED
//**
//** US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION OR
//** DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH IBM.
//********************************************************************
//STEP1 EXEC PGM=ANLSTPR,PARM='ssid,JOHNDOE'
//STEPLIB DD DISP=SHR,
// DSN=SYSX.ANL420.SANLOAD
// DD DISP=SHR, DSN=SYS1.CEE.SCEERUN
// DD DISP=SHR, DSN=SYS1.DSN##.SDSNEXIT
// DD DISP=SHR, DSN=SYS1.DSN##.SDSNLOAD
//DSNTRACE DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//QUERYIN DD DISP=SHR,  
// DSN=SYSX.ANL420.SANSQL(ANLTEST)
```

The QUERYIN DD statement points to the file containing the SQL statements that you want to analyze. If the test program was not bound properly before running, an SQL -805 error code could result.

For each SQL statement in your test input file, the test program ANLSTPR lists the SQL statement, the SQL error code, the DB2 SQL Performance Analyzer warning flags, and the DB2 SQL Performance Analyzer estimates for elapsed time, CPU
time, I/O count, QUNITS, and cost, as shown in the following figure.

```
* RRS IDENTIFY RETURNS  0  0  0
* SIGNON USER RETURNS  0  0  0
* CREATE THREAD RETURNS  0  0  0
* EXPLAIN PLAN FOR (LENGTH 240)
  * UPDATE SYSIBM.SYSTABLES
  * SET NPAGES = -1
  * WHERE NPAGES = -1

* ANLPRCR SQLCODE IS  0
* ANLPRCR RETURNS ===== WARNING FLAGS: -----  
  ELAPSED:  38.70953 CPU TIME:  0.27816
  I/O COUNT:  101 QUNITS:  16
  MONETARY:  1.18
  ANL CODE:  0 SQL CODE:  0

* EXPLAIN PLAN FOR (LENGTH 320)
* SELECT SUM(NTABLES), AVG(PARTITIONS) FROM SYSIBM.SYSTABLESPACE
* WHERE CREATOR <> 'SYSIBM'
* AND NTABLES > 1 AND
* SEGSIZE = 0

* ANLPRCR SQLCODE IS  0
* ANLPRCR RETURNS ===== WARNING FLAGS: ----- 
  ELAPSED:  1.09482 CPU TIME:  0.08612
  I/O COUNT:  18 QUNITS:  5
  MONETARY:  0.20
  ANL CODE:  0 SQL CODE:  0

* EXPLAIN PLAN FOR (LENGTH 640)
* SELECT A.CREATOR, A.NAME, B.COLNAME, B.ORDERING, C.COLSEQ
* FROM SYSIBM.SYSINDEXES A, SYSIBM.SYSKEYS B, SYSIBM.SYSFOREIGNKEYS C
* WHERE A.CREATOR = B.IXCREATOR
* AND A.NAME = B.IXNAME
* AND A.NAME = C.TBNAME
* AND A.CREATOR = C.CREATOR
* AND A.COLCOUNT > 1
* ORDER BY C.COLSEQ

* ANLPRCR SQLCODE IS  0
* ANLPRCR RETURNS ===== WARNING FLAGS: YY-YY 
  ELAPSED:  5223.66996 CPU TIME:  3403.41581
  I/O COUNT: 112 QUNITS:  185853
  MONETARY:  771.94
  ANL CODE:  0 SQL CODE:  0

* RRS TERMINATE IDENTIFY  0  0  0
* PROGRAM TERMINATION
```

* Figure 46. ANLSTPR sample program output

## Easy Explain report levels

You can use report levels to define the amount of information you want to see in your Easy Explain reports.

You can use a summary report to get a general idea of the state of SQL statements in a large plan. Easy Explain operating time might be considerable if a large plan with several hundred SQL statements is being investigated.
To get an idea of the state of the SQL statements in this large plan, it is recommended that you produce only a summary page as the first run. Use the LEVEL statement to produce a summary report, as shown in the following example:

```
Level=Summary
Plan=ANLITST
```

The one page report example is shown in the following figure, where the access path of each statement is given in one line for each miniplan.

```
1 SQL, QNO, PLN, and QMF requests were processed:

1: Plan=ANLITST

  DBRM/Pack Stmt Typ Plan owner: DPGROTH Bind date: 1992-01-18
  ANLITST 3 D L Matching index scan (1/2) - data pages scan
  ANLITST 3 D Additional Sort for ORDER BY
  ANLITST 3 D S* Table space scan - no index will be used
  ANLITST 32 D S* Table space scan - no index will be used
  ANLITST 32 D L Matching index scan (1/2) - data pages scan
  ANLITST 32 D Additional Sort for ORDER BY

```

If this report does not provide sufficient information and you want a fast run time, the next step could be to produce a report that contains the explain and access path information, but excludes index, table, and table space information. Such a report is produced using the following combination of statements:

```
Level=NoCatalog
Plan=LargeOne
```

Each explainable SQL statement in the plan is a separate page in the report.

For table and table space information, without index information, specify the following statement combination:
This combination of statements results in a fast response time because the non-indexed system catalog table SYSIBM.SYSKEYS is not accessed.

To retrieve all catalog information, including key distribution and information for all indexes on tables, where the investigated SQL statements perform table space scans, specify the following combination of statements:

```
Level=Detail
Plan=LargeOne,index=No
```

### EE�크 PATH table

The EE�크 PATH table is populated using Easy Explain and contains the access plans for Easy Explain queries.

This table joins other Easy Explain DB2 tables (EEPLAN, EEDEBRM) as data repositories held and managed by Easy Explain. Periodically, Easy Explain maintenance deletes old records from all these repositories. To manage your own access plans, you must create a view and maintain your own copy of this table.

The table contains the essential PLAN_TABLE and DSN_STATEMENT_TABLE cost items relative to the access path. It is used to generate the old access path cost estimates, and the old access plan itself. Easy Explain provides SQL statements from its data sources, such as the Catalog for plan or package statements, and stores the QUERYNO, APPLNAME, and TIMESTAMP to tie the SQL from user_ID.ANLEEE.SQL into these saved records.

DB2 SQL Performance Analyzer can take this existing plan data in EE�크 PATH, perform its access path cost analysis, and use the SQL from userid.ANLEEE.SQL to prepare a new cost analysis and plan. SQL PA then prepares a report that shows the OLD plan and cost, the NEW plan and cost, and a comparison of the two. For each user, authorization_ID.EEPath is the name of this table. DB2 SQL Performance Analyzer uses the USEPLAN parameter to select the authorization_ID qualifier.

The QUERYNO is passed with the SQL statements that correspond to the plan records captured in the EEPath table. The Application Name and Timestamp help to further identify the proper records for consideration and analysis.

### SQL data file: user_ID.ANLEEE.SQL

The user_ID.ANLEEE.SQL data file is the Easy Explain input source.

The SQL extracted from any Easy Explain data source (for example, a Plan, Package, QMF, Source Program, or File Input) is passed to DB2 SQL Performance Analyzer in a standard fixed block 80 byte record file, within cc 1-72 (leaving 73-80 for line numbers, if any). It uses a standard name to differentiate it as the Easy Explain input source: user_ID.ANLEEE.SQL, qualified by the TSO user ID which is the authorization ID that is running the programs. You can set up this file with a different prefix.

### TOPA parameter

Easy Explain uses a special parameter, TOPA, to force SQL and access plans to the EEPath table and user_ID.ANLEEE.SQL files.
TOPA=ALL setting is limited to use with plans and packages, but it can also be set to YES (default) or NO for other Easy Explain input sources. When set to ALL, all SQL extracted by Easy Explain from the catalog plans and packages is saved for later analysis by SQL PA, including old versus new cost comparisons.

TOPA=YES has certain excessive query criteria already in place within Easy Explain that forces SQL to be written if any query exceeds a 10 MSU cost estimate, contains star joins, or contains other complex SQL structures.

**EXPLAIN parameters**

The DB2 SQL Performance Analyzer user parameters, EXPLAIN, QUERYNO/STOPQNO, EEECALL, and EEEPATH, help the interface between SQL PA and Easy Explain work efficiently.

They are used in the ANLPARM file for the DB2 SQL Performance Analyzer part of the batch run.

**VIADRDA remote connect**

SQL PA can process a remote copy of the product running under another DB2 subsystem, on the same server or another server, by using the DRDA protocol and the SQL CONNECT statement.

If the VIADRDA parameter is specified for DB2 SQL Performance Analyzer, SQL requests are routed to other copies of the DB2 SQL Performance Analyzer programs on remote systems. Define the VIADRDA parameter in the ANLPARM file, or on the Change System Parameters panel under TSO ISPF. Setting the parameter to +OFF+ turns off the feature. The TSO parameter remains set to the default (+OFF+) value unless the TSO user wants to route processing to another DB2 subsystem. These users might experience some delay in response time if the dispatching priority of DDF requests on the remote system is not as high as the local DB2.

VIADRDA allows routing of your SQL request to other copies of DB2 SQL Performance Analyzer, while running on a local system. To use TSO to explain a remote package, set the VIADRDA parameter and specify the Remote location field on the Process Packages panel.

The format of the VIADRDA parameter is:

**VIADRDA location-name | +OFF+**

Where location-name, a 16-character field, is any valid location specified in the local SYSIBM.LOCATIONS table. You can leave the parameter in ANLPARMs with a value of +OFF+ to make it non-operational.

Because of the size of the parameter field (16 characters), the comments for this parameter must start in column 31.

- **REPORTS ALL**
- **VIADRDA DALLAS-CQ**
- **DBRMKEY +OFF+**

The remote locations must contain fully installed versions of DB2 SQL Performance Analyzer, including a Registry table, and PLAN_TABLE. Then, users can run DB2 SQL Performance Analyzer, their SQL, and DB2 subsystem, through the DRDA Connect facilities.
After connection to another DB2 subsystem, the entire contents of the ANLIN file are processed under that remote system, using the DB2 SQL Performance Analyzer program installed there. Results are returned to the local system.

Topics:

• "Bind considerations"

## Bind considerations

The following information includes bind considerations for the DRDA protocol.

Under the target (local) host system, the bind directive is amended to include a Package List and ensure the DRDA protocol. Amendments are in bold type:

```
BIND PLAN (ANLSQLPA) MEMBER(ANLSQLPA) ISOLATION (CS) –
   ACTION(REP) VALIDATE (BIND) CACHESIZE(4096) RETAIN® –
   QUALIFIER(ANLUSER1) DYNAMICRULES(RUN) DBPROTOCOL(DRDA) –
   PKLIST (location.SQLPA.ANLSQLPA)
```

Each location to be accessed remotely must be separately included in the PKLIST.

Under the remote systems, the plan is also bound as a package so that it can be called:

```
BIND PACKAGE (location.SQLPA) MEMBER(ANLSQLPA) ACTION(ADD) –
   ISOLATION(CS) VALIDATE(BIND) QUALIFIER(ANLUSER1) –
   DYNAMICRULES(RUN) DBPROTOCOL(DRDA)
```

In both cases, the `location` must refer to a location name stored in the SYSIBM.LOCATIONS table of the local system and referenced in the VIADRDA parameter.
Chapter 4. DB2 SQL Performance Analyzer run parameters

There are two sets of parameters used by SQL PA.

One set of parameters is user-oriented and can be found in $hiqual$.SANLPARM member $ssid$PARM.

The second set of parameters describes the configuration of the production system where the application being processed by DB2 SQL Performance Analyzer actually runs, and can be found in $hiqual$.SANLPARM member $ssid$CNTL. The $ssid$CNTL parameters describe the configuration of the host system. The CNTL parameters include the type and speed of the CPU processor, the DB2 buffer pool sizes, and other systems-oriented information. Configuration parameters also include the costing guidelines and excessive use limits which trigger warning messages from DB2 SQL Performance Analyzer. The best practice is to implement the parameters consistently and uniformly for every user of DB2 SQL Performance Analyzer.

You can also use the ISPF interface to modify the DB2 SQL Performance Analyzer parameters by using Modify Configuration panel option of Change SQL PA parameters.

Topics:
- “Impact on cost estimates”
- “Parameter file characteristics” on page 145
- “Specifying ANLPARM user parameters” on page 145
- “Specifying threshold control parameters” on page 146
- “Specifying ANLCNTL configuration parameters and selecting the target host system” on page 147
- “Enabling the predicate analysis report” on page 148
- “Optimizing the table and SQL report processing” on page 148
- “The difference between SANLDATA and SANLPARM” on page 149
- “Qualifying unqualified objects in plans and packages” on page 149
- “Qualifying unqualified objects in DBRMs and flat files” on page 150

Impact on cost estimates

Most of the parameters specified in the ANLPARM and ANLCNTL parameter sets have some impact on the internal formulas and costing algorithms used by SQL PA.

The following list reviews the impact that specific parameters have on cost estimates.

ANLKEYS
Alterns consideration of certain path lengths and instruction counts, leaving out some processor usage depending upon option selected.

BUFF08K
Has equivalent effect of BUFFERS on 8 KB page applications.

BUFF16K
Has equivalent effect of BUFFERS on 16 KB page applications.
BUFF32K
Has equivalent effect of BUFFERS on 32 KB page applications.

BUFFERS
Influences the number of pages contained in a prefetch block for user requests from 4 KB buffer pools.

BUFFHIT
Governs the logical-to-physical I/O ratio, allocating a portion of the total logical I/O to disk; impacts the I/O waits, elapsed time, and other calculations.

CONNECT
Imposes processor usage charges for thread management, attach facility and application logic, different for each connection chosen.

CPUCOST
Establishes the base for the monetary cost estimates of CPU time.

DATASHR and DSGROUP
Influences the processor usage applied to data sharing in a Sysplex environment, and the overall processing capacity of the system.

DEGREES
Enables the optimizer to consider parallel processing; impacting sequential prefetch, I/O wait, and elapsed time for table scans.

DYNAMIC
Influences the processor usage for dynamic SQL processing.

ENGINES
Contributes to determining the speed of a single processing engine, the fastest that any instructions start.

ESACOMP
Invokes different path length algorithms within SQL PA for table spaces using the hardware-assisted data compression facility.

ESASORT
Invokes different path length algorithms within SQL PA for systems with the hardware-assisted DB2 sort facility.

IOSCOST
Establishes the base for the monetary cost estimates of physical I/O.

HPOOLRD
Governs read time of physical I/O: are they found in expanded storage, or read from disk? Impacts elapsed times and I/O waits; also influences asynchronous writes back to pool after insert, update, or delete activity.

LPARENG
Helps rate individual processor speed on processor complexes with multiple system images.

NEWSTOR, NEWSEEK, NEWROTA, and NEWXFER
Redefine the basis for service times calculated for all physical I/O calls, based on the seek, revolution speed and transfer rate of the new devices. The parameters also impact the wait time estimates and contribute to overall elapsed time.
PRECISE
Causes SQL PA's calculated CPU times and QUNITS to be replaced with estimates from the DB2 optimizer, and only CPU time is considered in SQL PA's costs when PRECISE is YES.

REFRESH
Allows the optimizer to consider materialized query tables (MQTs) that are summaries, aggregates, or reduced row versions of the base tables as access path candidates.

STORAGE
Provides the basis for service times calculated for all physical I/O calls, based on the seek, revolution speed and transfer rate of the devices. This parameter also impacts the wait time estimates and contributes to overall elapsed time.

TIMCOST
Establishes the base for the monetary cost estimates of elapsed time.

VERSION
Selects which set of path lengths are used, with different instruction counts on some functions.

You can obtain many varied results by manipulating the SQL PA parameters. It is important to remember that SQL PA can reflect any existing or proposed configuration description in its parameter set.

Parameter file characteristics
All DB2 SQL Performance Analyzer parameters, when stored in files, have certain characteristics in common.

The common characteristics for SQL PA parameters include the following characteristics:
• Each parameter keyword begins in column 1, is exactly seven characters long, and is followed by a blank.
• You can add comments to any parameter after column 20 except for those parameters whose value can be longer than 10 characters.
• The maximum length of a comment is 60 characters.
• Non-line numbered files are preferred and fixed block 80 byte records are expected.

Specifying ANLPARAM user parameters
You can change DB2 SQL Performance Analyzer user parameters to get results that are tailored to your specific requirements.

Before you begin
Create a backup copy of the hiqual.SANLPARAM member to preserve the original settings.

For more information about the ANLPARAM parameters, see "ANLPARAM user parameters" on page 267.
**Procedure**

Use one of the following methods to change the ANLPARM parameters:

- Use ISPF to change the ANLPARM parameters by editing the `high_level_qualifier.SANLPARM` member `ssidPARM` and change the parameter values as necessary.
- Use SQL PA to change the ANLPARM parameters by completing the following steps:
  1. On the SQL PA main menu, specify option 3, Modify DB2 SQL PA Configuration. The Modify Configuration panel is displayed.
  2. Specify option 3, Change the DB2 SQL PA parameters. The Parameter Data Sets panel is displayed.
  3. Optional: If you selected (USRPARM) Allow users to have a copy of the user parameters on the Product Parameters panel in Tools Customizer, specify a data set name in the User-level parms field. The user parameters from the data set that is specified in the SQL PA user parms field are copied into the data set that is specified in the User-level parms field. If this data set does not exist, it will be created.
  4. In the Edit SQL PA user parms field, specify YES, and press Enter. The User Parameters panel is displayed.
  5. Modify any Environment Settings parameters, and press Enter.
  7. Modify any Easy Explain parameters, and press Enter.
  8. Press Enter to save the settings.

**Specifying threshold control parameters**

Use the threshold control parameters to control the levels at which certain SQL Advisor messages are issued.

**About this task**

You can turn off certain messages and capture only the message ID, excluding the text, using threshold control parameters.

**Procedure**

1. Determine which of the threshold parameters you want to use for your environment. For detailed information on each of the parameters, see “ANLPARM user parameters” on page 267.
2. Modify one or more of the following threshold parameters by using the ISPF or the ANLPARM file.
   - ALLPART YES | NO
   - INLISTS nnn
   - ISCANPG nnnnnnn
   - IXUPDAT nnn
   - JOINTAB nnn
   - MATCOLS n.nn
   - MESSAGE YES | NO
   - NONIXPG nnnnnnn
   - NONONIX YES | NO
Specifying ANLCNTL configuration parameters and selecting the target host system

DB2 SQL Performance Analyzer has the capability to project its anticipated cost and performance estimates on any configuration, including production host machines and hypothetical configurations not yet in place. The ANLCNTL parameters are not normally modified by DB2 SQL Performance Analyzer users.

**About this task**

The parameters define a target host configuration.

You can run DB2 SQL Performance Analyzer on a test machine and to forecast application performance on a production system.

You can have multiple configurations (PDS members) in batch using the ANLCNTL DD statement to point to the `hiqual.SANLDATA` PDS member, and in TSO by having multiple members in the `hiqual.SANLPARM` PDS.

The ANLCNTL system configuration parameters are stored in a PDS library, allowing you to select a specific target system (PDS member) for the cost analysis of SQL PA.

For SQL PA operating under TSO online, the `hiqual.SANLPARM` library is provided as a place to store the various configurations for different host systems, including ones that exist at your site and sample configurations that you want to test against. This PDS is populated with members for each DB2 host during installation.

In batch mode you can allocate one of these members in the ANLCNTL DD statement, point to your own copy as a flat file, or use SYSIN DD * and define them in-stream.

**Procedure**

1. Determine which of the configuration parameters you want to use for your environment. For detailed information about each of the parameters, see “ANLCNTL configuration parameters” on page 262.
2. For each member (subsystem, LPAR, or configuration), edit the member for ANLCNTL configuration parameters or use ISPF. To use ISPF to edit the parameters, use the following steps:
   a. From the primary menu, select Modify DB2 SQL PA Configuration.
   b. Select Change the DB2 SQL PA parameters.
   c. Specify Yes on Edit SQL PA system parms.
   d. Modify and save the new parameter values.
Enabling the predicate analysis report

Enabling this feature is optional, but if it is turned on, the extra predicate analysis report is generated. If it is not activated, the PRECISE ALL parameter does not produce a predicate analysis report.

About this task

The ALL setting provides more precise information in evaluating SQL for DB2, including a Predicate Analysis section in the Explain and Detailed Trace reports.

When PRECISE is set to YES, DB2 SQL Performance Analyzer attempts to replace its path length or instruction count estimates (the CPU time, essentially) with the internal estimates from the DB2 optimizer, drawn from the Explain table DSN_STATEMNT_TABLE.

The QMF Interface for DB2 SQL Performance Analyzer and the DB2 SQL Performance Analyzer stored procedure can also use the optional Explain table (DSN_STATEMNT_TABLE) with the PRECISE YES parameter.

The default for this parameter is YES.

The option of using the optimizer’s own internal cost estimate (path length) to process each SQL request is available in all the DB2 SQL Performance Analyzer environments—batch, TSO, QMF, and as a DB2 stored procedure. This ensures the highest available level of accuracy attainable by DB2 SQL Performance Analyzer users from any environment.

Procedure

To generate a predicate analysis report in DB2 use the following procedure.

1. Edit the member for ANLCNTL configuration parameters or use ISPF. To use ISPF to edit the PRECISE parameter, use the following steps:
   a. From the primary menu, select Modify DB2 SQL PA Configuration.
   b. Select Change the DB2 SQL PA parameters.
   c. Specify Yes on Edit SQL PA user parms.
   d. Modify and save the new parameter value.
2. Run the ssidJSPA job using the PRECISE ALL parameter. The predicate analysis report is also available under TSO online.

Optimizing the table and SQL report processing

You can optimize table and SQL report processing by bypassing the Easy Explain function.

About this task

To retrieve table report information without having to run EXPLAIN on every statement, set the FASTTBL parameter to YES.

FASTTBL=NO shows information about the shared and exclusive DB2 locks, and the SEL, DEL, INS, UPD, and MER columns reflect query blocks, LIS and LIX reflect DB2 locks, and STM reflects SQL statements (not query blocks).
Procedure
1. Review the FASTTBL parameter settings. For detailed information, see "ANLCNTL configuration parameters" on page 262.
2. Edit the member for ANLCNTL configuration parameters or use ISPF. To use ISPF to edit the FASTTBL parameter, use the following steps:
   a. From the primary menu, select Modify DB2 SQL PA Configuration.
   b. Select Change the DB2 SQL PA parameters.
   c. Specify Yes on Edit SQL PA system parms.
   d. Modify and save the new parameter values.

The difference between SANLDATA and SANLPARM
The DB2 SQL Performance Analyzer ISPF interface is intended to use the hiqual.SANLPARM library for its parameter values, leaving the hiqual.SANLDATA library available for different values that can be used for the batch interface.

Either interface can specify the members from either library. However, there are certain parameter values that are only valid for batch use and are ignored when using ISPF.

When setting up the DB2 SQL Performance Analyzer stored procedure, a copy of the SANLPARM member ssidCNTL is recommended. The copy can be allocated to the allocated to the stored procedure address space so that it does not prevent other users from accessing those parameters.

Qualifying unqualified objects in plans and packages
Occasionally, you need to explain statements from plans and packages that have unqualified objects. Objects are qualified differently depending on the interface you are using.

About this task
For ISPF interfaces, the SQL ID that displays on the panels qualifies your unqualified objects. Alternatively, you can use the DB2 supplied stored procedures, DSN8EXP or DSN8EXP, to set the qualifier.

For batch interfaces, the QUALIFY user parameter is what qualifies your unqualified objects. For more information about using the QUALIFY parameter, see "ANLPARM user parameters" on page 267.

Procedure
1. For ISPF interfaces, define the qualifier used to qualify unqualified objects in one of the following ways:
   • Use the ISPF interface to change the SQL ID by modifying the configuration (option 3 from the primary menu) and then changing the current DB2 SQLID (option 2).
   • Use the DB2 supplied stored procedures to set default qualifiers for DB2 SQL Performance Analyzer and Easy Explain. For both DB2 SQL Performance Analyzer and Easy Explain, specify YES for the DSN8EXP parameter in the ANLPARM user parameter file.
2. For batch interfaces, define the QUALIFY parameter in the ANLPARM user parameter file.
Qualifying unqualified objects in DBRMs and flat files

When explaining statements from DBRMs and flat files that have unqualified objects, change the qualification using the ISPF interface or the batch interface.

About this task

Occasionally, you need to explain statements from DBRMs and flat files that have unqualified objects. Objects are qualified differently depending on the interface you are using.

For ISPF interfaces, there is a table qualifier field on the panel. The field is used to qualify the tables, but if it is blank, the SQL ID displayed on the panel is used. Only one value for this field is allowed.

For batch interfaces, the QUALIFY user parameter is what qualifies your unqualified objects. For more information about using the QUALIFY parameter, see “ANLPARM user parameters” on page 267.

Procedure

1. For ISPF interfaces, change the SQL ID by modifying the configuration (option 3 from the primary menu) and then changing the current DB2 SQLID (option 2).
2. For batch interfaces, define the QUALIFY parameter in the ANLPARM user parameter file.
Chapter 5. Easy Explain input specifications

You can use Easy Explain input commands to set the parameters that SQL PA uses for analysis.

Topics:
- “Easy Explain process”
- “Configuring Easy Explain”
- “Easy Explain plan table” on page 152
- “Easy Explain input command requirements” on page 152
- “Easy Explain input commands” on page 153
- “Input command definitions” on page 160
- “Subparameter definitions” on page 163
- “Specifying language and explaining data from another user” on page 170
- “Selected authorization identification” on page 171
- “User authorization table” on page 172
- “User defaults data set” on page 172
- “Transferring complex queries to SQL PA” on page 173

Easy Explain process

Easy Explain performs several checks and activities when it runs.

Easy Explain checks to make sure that all necessary tables are accessible.

If all necessary tables are accessible, Easy Explain reads the first input command. The command is checked for validity, and one of the main processes is started.

Each input specification produces a complete report. Each report can consist of several pages.

If LEVEL=SUMmary is in effect, information for the current SQL statement is shown only in the summary page.

At normal completion of Easy Explain, the last page shows a listing of all the input specifications with prefixed report numbers. You can use this page as a table of contents.

Configuring Easy Explain

Easy Explain makes extensive use of DB2 system catalog tables.

About this task

To run Easy Explain you must define several settings.

Procedure

1. Define SELECT authorization to the following catalog tables:
   - SYSIBM.SYSDATABASE
SYSIBM.SYSTABLESPACE
SYSIBM.SYSTABLES
SYSIBM.SYSCOLUMNS
SYSIBM.SYSINDEXES
SYSIBM.SYSFIELDS
SYSIBM.SYSKEYS
SYSIBM.SYSTMT
SYSIBM.SYSPLAN
SYSIBM.SYSDBRM
SYSIBM.SYSPACKLIST
SYSIBM.SYSPACKAGE
SYSIBM.SYSPACKSTMT
SYSIBM.SYSCOLDIST

2. If you expect to run EXPLAIN on saved QMF queries, the Easy Explain plan
   binder must have SELECT authorization to the following QMF tables:
   • Q.OBJECT_DIRECTORY
   • Q.OBJECT_DATA

3. To control user access to the catalog tables, Easy Explain selects data through
   views or synonyms. Use the DDL files supplied with Easy Explain or customize
   them to fit your installation standards.

4. If you do not want the current user's PLAN_TABLE table to be created in the
   default database, create a PLAN_TABLE table in a specified database and table
   space before you run Easy Explain with an SQL or QMF statement.

Easy Explain plan table

Easy Explain uses a PLAN_TABLE either prefixed with the authorization ID of the
job submitter or belonging to the binder of the plan being explained.

For each SQL and QMF command, Easy Explain verifies that a PLAN_TABLE
exists for the current authorization ID. Easy Explain also verifies if a PLAN_TABLE
exists if a dynamic EXPLAIN is to be issued for a DML SQL statement in a plan or
package.

If you have SELECT authorization to the PLAN_TABLE of another user, you can
select EXPLAIN information from this table using one of the following methods:

• You can specify the authorization ID of another user in the PARMS field of the
  DB2 RUN statement.
• You can specify the authorization ID in the OWNER parameter of the QNO
  statement.

If a PLAN_TABLE does not exist, and you are doing a dynamic explain using the
SQL or QMF command, and you have a CREATETAB authorization, Easy Explain
creates a PLAN_TABLE in the default database. If the installation has specified a
database and table space name in the location view of the EEEWORK table, the
dynamically created PLAN_TABLE tables are created in the specified location.

If you are doing a dynamic EXPLAIN using the SQL or QMF command, Easy
Explain determines that columns need to be added to the table, and adds them.

Easy Explain input command requirements

The rules you must follow when using Easy Explain input commands are
explained.
To allow the parser to work, your input specifications must adhere to the following rules:

- Start the Easy Explain input command with one of the following statements:
  - Comment
  - Level
  - Package
  - Plan
  - QMF
  - QNO
  - SET
  - SQL
  - TOPA

- Specify only one keyword statement per line.

- Start the keyword statements in position 1, and must immediately be followed by an equal sign (=). Further, the input parameter must immediately follow the equal sign (no embedded blanks). The format is free after the input parameter. You can specify keywords, Plan and QMF names, and SQL statements in both upper and lower case.

- An SQL statement can span several lines.

- End certain statements with a semicolon (;). If you are using the SQL= keyword, you must end the SQL statement with a semicolon (;), if it is followed by another Easy Explain statement. You can also optionally stop the LEVEL, SET, QNO, QMF, PLAN, and PACKAGE commands with a semicolon (;).

- Generate a Summary report, you must specify a level statement, such as LEVEL=SUMmary.

**Easy Explain input commands**

Easy Explain reads input commands from a data set with the ddname INPUT. The following information describes detailed command specifications.

**Comment statement**

```
---Any text as comment (shown on Summary Page)
```

```
---Any text as comment (not shown on Summary Page)
```

If the first two characters of a command line are hyphens (--) or the first character of a command line is an asterisk (*), the entire command line is regarded as a comment. In the first case, the comment is printed on the Summary Page, but if the first character is an asterisk (*), the comment is not shown in the output.
Level statement

The LEVEL command controls how much information is shown in reports or other output. The command must have one of the following two formats:

- `Level=xxx`
- `LVL=xxx`

DETail is the default value. For more information about the acceptable values, see “Input command definitions” on page 160.

Note: A LEVEL specification is in effect until a new LEVEL value is specified.

SET statement

The degree of parallelism for a dynamic SQL (or QMF) statement is set by setting the current degree to ANY or 1 (default). The setting is in effect until a new SET CURRENT DEGREE statement is issued.
The setting of a current SQL ID is done using the same statement as in standard SQL DML. If SET CURRENT SQLID=USER is specified, the primary SQL ID is set; otherwise, a valid secondary SQL ID is set. This setting is in effect until a new SET CURRENT SQLID statement is issued.

**Note:** For compatibility with prior releases of Easy Explain, the SQLID= or SID= statement is still supported (SQLID=SecUser).

---

**Current optimization hint**

```
SET CURRENT OPTIMIZATION HINT = 'Hint String', ID = number
```

---

DB2 hints are passed to the Optimizer through the use of the SET CURRENT statement. In the top part of the Easy Explain report, Easy Explain displays an ID number for the current miniplan:

```
Opthint: , Hint: , Dir.Row: No , OptHint-ID: 1321
```

The number 1321 in the following example uniquely identifies a miniplan, where a DB2 hint is given. If you decide to supply a DB2 hint for a given miniplan, you can issue a SET CURRENT HINT command, where the ID-number is supplied together with a hint string, such as:

```
SET CURRENT HINT = 'Myhint', id=1321
```

---

**TOPA Statement**

```
TOPA= No
```

---

Before explaining an SQL statement or a QMF query, you can specify if detailed explain information can be collected for later transmission to DB2 SQL Performance Analyzer. If you want the detailed information, specify the following: TOPA=Yes

This specification is in effect for subsequent SQL= and QMF= statements until TOPA=No is specified or the program stops. For information about JCL requirements, refer to "Transferring complex queries to SQL PA" on page 173.

---

**SQL Statement**

```
SQL= Explainable SQL Statement (several lines);
```
To run Easy Explain on an SQL statement, the SQL command must have the following format:

\[\text{SQL}=\text{xxx}\;;\]

The statement can span several lines and can include comments (indicated by '--'). The statement must be ended by a semicolon (;), if it is followed by another Easy Explain statement.

**QMF Statement**

To run Easy Explain on a QMF query, the QMF command must have the following format, where \text{xxx} is an optional authorization ID of the query owner. If \text{xxx} is not specified, the query is assumed to be owned by the job submitter. The name of the QMF SQL query is \text{yyy}.

\[\text{QMF}=\{\text{xxx}\.\}\text{yyyzzz}\]

The valid values for the INDEX subparameter are described in “Subparameter definitions” on page 163.
To run Easy Explain on a plan, the PLN command must have one of the following two formats, where \( a \) is the name of the plan to be explained:

\[
\text{Plan} = a, [\text{dbrm}=b], [\text{index}=c], [\text{first}=d], [\text{last}=e], [\text{table}=f], [\text{acctype}=g], [\text{format}=h], [\text{level}=i], [\text{packages}=j], [\text{gen}=k], [\text{hostvar}=l], [\text{location}=m], [\text{planinfo}=n], [\text{topa}=o]
\]

\[
\text{PLN} = a, [\text{dbrm}=b], [\text{index}=c], [\text{first}=d], [\text{last}=e], [\text{table}=f], [\text{acctype}=g], [\text{format}=h], [\text{level}=i], [\text{packages}=j], [\text{gen}=k], [\text{hostvar}=l], [\text{location}=m], [\text{planinfo}=n], [\text{topa}=o]
\]

The valid values for the subparameters are described in "Subparameter definitions" on page 163.
To run Easy Explain on a package, the PKG command must have one of the following two formats, where \( a \) is the combination of the collection ID, the package ID, and the version ID:

\[
\text{PKG=}a[,\text{index=c}][,\text{first=d}][,\text{last=e}][,\text{table=f}][,\text{acctype=g}]\[,\text{format=h}][,\text{level=i}][,\text{force=j}][,\text{gen=k}][,\text{hostvar=l}]\[,\text{location=m}][,\text{topa=n}]
\]

The valid values for the subparameters are described in "Subparameter definitions" on page 163.
To run Easy Explain on a query number, the QNO command must have one of the following two formats, where \( xxx \) is an existing query number in the \( \text{PLAN\_TABLE} \) that is being accessed:

\[
\begin{align*}
\text{QNO=} & \ xxx \ [,\text{owner=}yyy][,\text{index=}zzz] \\
\text{QNO=} & \ xxx \ [,\text{owner=}yyy][,\text{index=}zzz]
\end{align*}
\]

The \( \text{OWNER} \) subparameter \( yyy \) is the authorization ID of the \( \text{PLAN\_TABLE} \). See "Subparameter definitions" on page 163 for a description of the \( \text{OWNER} \) subparameter. For additional information see "Selected authorization identification" on page 171.

**Valid command examples**

The following table displays some examples of valid commands.

**Table 13. Valid command examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>QNO=1234,owner=XYZ</td>
<td>QUERYNO. 1234 in XYZ.PLAN_TABLE is processed</td>
</tr>
<tr>
<td>plan=myplan;</td>
<td>All explained statements in MYPLAN are processed</td>
</tr>
<tr>
<td>LVL=summary</td>
<td>Produces summary information only</td>
</tr>
<tr>
<td>QmF=user01.little_query,index=T</td>
<td>QMF query user01.little_query is processed. If table space scan is used, detailed index information for all indexes of the accessed table are shown. Otherwise, only information for the chosen index is shown</td>
</tr>
<tr>
<td>level=detail</td>
<td>Back to detailed information</td>
</tr>
<tr>
<td>SQL=select A, B, C from USER1 .TABLE3 where A&gt;'abc' and B=78</td>
<td>The specified SQL query is processed</td>
</tr>
</tbody>
</table>

**Invalid command examples**

Some examples of invalid commands are shown in the following table.

**Table 14. Invalid command examples**

<table>
<thead>
<tr>
<th>Command</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>qno= 2345;</td>
<td>A space after the equals sign (=) is not permitted</td>
</tr>
</tbody>
</table>
Table 14. Invalid command examples (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>QNO=ABCD</td>
<td>The query number is not numeric</td>
</tr>
<tr>
<td>sql=SELECT * FROM DSN8610.EMP</td>
<td>The SQL statement must end with a semicolon ($)</td>
</tr>
<tr>
<td>Qmf=ABC.MyQuery</td>
<td></td>
</tr>
</tbody>
</table>

Input command definitions

The following information defines the input command statements and their valid values.

Comment

If the first two characters of a command line are hyphens (--), or the first character of a command line is an asterisk (*), the entire command line is regarded as a comment. In the first case, the comment is printed on the summary page, but if the first character is an asterisk (*), the comment is not shown in the output.

LEVEL

This function is used to control the level of detail. You can use Level or LVL as the keyword. By default, the report level is detail.

Use the LEVEL=keyword parameter to define the level of detail you want in reports. Alternatively, you can specify a LEVEL subparameter on the PLAN-statement to set the report level for the specified plan.

ACCESSpath

When this parameter is specified, only the access path, table, and index are shown.

ALLIndex

When this parameter is specified, all indexes for the accessed table are shown.

DETail

Any parameter that is not SUMMARY, NOKeys, SQLsummary, and NOCatalog is regarded as DETail. Specifying this parameter results in a full detailed report for each SQL statement. DET is the default value for the Level parameter.

KEYdist

When this parameter is specified, the distribution of the 10 most used key values is shown. For all other information, the level is regarded as Detail.

NOCatalog

When this parameter is specified, only EXPLAIN data and the Access Path are shown. It does not produce index, key, table, table space, and bind information.

NOKeys

When this parameter is specified, all key field information is excluded from the report.

SQLsummary

When this parameter is specified, only the SQL statement and Access Path messages are shown.
SUMmary
When this parameter is specified, only the summary page with one line for each SQL statement is produced.

TABular
When this parameter is specified, the SQL statement and a brief report is displayed. Plan table information displays in tabular form followed by statistical information.

CONDensed
When this parameter is specified, the SQL statement and a condensed report is displayed. Plan table information displays in a condensed form that is similar to the tabular form followed by statistical information.

TIYes /TINo
These parameters toggle the printing of the table and index names for each statement in the summary page.

XON / XOFF
These parameters toggle the printing of the explain data listed immediately after the SQL statement being explained.

PACKAGE
When a collection and package name are supplied as input, Easy Explain verifies that the specified package does exist in the catalog table SYSIBM.SYSPACKAGE. If the package does exist, Easy Explain verifies that the package was bound with the EXPLAIN option. If it is not bound with an EXPLAIN option, an EXPLAIN command is dynamically issued for the individual DML SQL statement. The output is stored in the PLAN_TABLE table of the job submitter with a query number of 999,999,998.

The name of the package creator is extracted from the SYSIBM.SYSPACKAGE table.

The EXPLAIN information is selected from the PLAN_TABLE owned by the package owner if the job submitter has select authorization to this table. If not, an error message is posted and Easy Explain continues with the next request.

A user ID that is specified in the PARMS field has no meaning for this function.

If all checks are positive, Easy Explain begins processing the lowest query number in the earliest version within the package. The program proceeds to the next query number and produces a new report for this query number. Processing continues until all statements in the package have been processed.

The run time for this function depends on the number of rows in the accessed PLAN_TABLE. Generally, you create two indexes to your PLAN_TABLE table, if it holds more than 400 to 500 rows. One index is built on the QUERYNO column, the other index is created on PROGNAME, QUERYNO, APPLNAME. Alternatively, you can delete unnecessary rows from your PLAN_TABLE and run RUNSTATS against the table space.
A large part of the run time is spent accessing the non-indexed table SYSIBM.SYSKEYS to provide index information. If this information is not essential, you can suppress access to the table by specifying INDEX=N after the package name.

**PLAN**

When a plan name is supplied as input, Easy Explain verifies that the specified plan does exist in the catalog table SYSIBM.SYSPLAN. If the plan exists, Easy Explain further verifies that the plan was bound with the EXPLAIN option. If this is not the case, an EXPLAIN command is dynamically issued for the individual DML SQL statement. The output is stored in the PLAN_TABLE table of the job submitter with a query number of 999,999,998.

The name of the plan creator is extracted from the SYSIBM.SYSPLAN table. The EXPLAIN information is selected from the PLAN_TABLE owned by the plan creator if the job submitter has select authorization to this table. If not, an error message is posted and Easy Explain continues with the next request.

The keyword for this function is Plan or PLN. A specified user ID in the PARMS field is meaningless for this function.

If all checks are positive, Easy Explain begins processing the lowest query number in the first DBRM within the plan. The program proceeds to the next query number and produces a new report for this query number. Processing continues until all statements in the plan have been processed.

The operating time for this function depends on the number of rows in the accessed PLAN_TABLE. Generally, you create two indexes to your PLAN_TABLE table if it holds more than 400 to 500 rows. One index is built on the QUERYNO column, the other index is created on PROGNAME, QUERYNO, APPLNAME. Alternatively, you can delete unnecessary rows from your PLAN_TABLE and run RUNSTATS against the table space.

A large part of the operating time is spent accessing the non-indexed table SYSIBM.SYSKEYS to provide index information. If this information is not essential, you can suppress table access by specifying INDEX=N after the plan name. Alternatively you can specify LEVEL=NOKeys on the PLN statement which excludes the key field information.

**QMF**

When using this function, you are able to EXPLAIN saved or generic QMF queries, providing the queries are written in the SQL language.

You must convert QBE and PROMPTED queries to SQL before they are explained.

Apart from explaining your saved QMF queries, you can EXPLAIN a query created by another user if the query was saved with SHARE=YES.

The QMF query can contain parameters (such as &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 ('). Also, the query must not contain single quotes in a comment.

If all tests are positive, processing continues the same as with the SQL keyword.

**QNO**

This function can be used when a dynamic EXPLAIN statement has been run with a given query number.
The EXPLAIN statement can either have been run from DB2I or QMF. Specify the query number as a parameter to the QNO keyword. Use QUERYNO as an alternative to the QNO keyword.

The specified query number might also be a statement number from a DB2 precompile list, if the application was bound with the EXPLAIN(YES) parameter on the BIND statement.

If a user ID was specified in the PARMS field, Easy Explain searches for the query number in the PLAN_TABLE that belongs to this user ID. If no PARMS field was specified, Easy Explain searches for the query number in the PLAN_TABLE of the job submitter.

If the specified query number does not exist in the PLAN_TABLE, a warning is posted and Easy Explain continues with the next command from the input data set.

**SET**  Setting of authorization ID, DEGREE, or hints.

**SQL**  When using this function, you supply an SQL DML statement as input to Easy Explain. The supplied statement is explained by Easy Explain, using a query number of 999,999,999. If this number exists in the PLAN_TABLE, Easy Explain deletes the rows before processing the SQL statement.

Specify the SQL statement to be investigated in the same way as in DB2I, that is, a free format using several lines with optionally embedded comments (starting with --). If the statement is followed by another Easy Explain statement, it must end with a semicolon (;), otherwise a semicolon is optional.

The EXPLAIN function is always run against the PLAN_TABLE of the job submitter. A user ID that is specified in the PARMS field has no meaning for this function.

After a successful EXPLAIN is run, the rows with the query number 999,999,999 are processed.

**TOPA**  Specify generation of output to transfer to DB2 SQL PA.

---

**Subparameter definitions**

The following information defines the subparameters and their valid values.

**ACCTYPE**  You can specify the ACCTYPE subparameter for the PLAN keyword to control the statements for explanation based on the chosen access path.

<table>
<thead>
<tr>
<th>Subparameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>This parameter is the default value. For each explainable SQL statement in the plan, a report is produced.</td>
</tr>
<tr>
<td>Matching</td>
<td>When this subparameter is specified, only explainable SQL statements, where an access path of matching index scan has been selected, are processed. After processing the results are shown in the Easy Explain report.</td>
</tr>
<tr>
<td>Nonmatch</td>
<td>When this subparameter is specified, only explainable SQL statements where an access path of non-matching index scan has been selected. After processing the results are shown in the Easy Explain report.</td>
</tr>
</tbody>
</table>
Tabscan
When this subparameter is specified, only explainable SQL statements where an access path of table space scan has been selected. After processing the results are shown in the Easy Explain report.

Hash
When this subparameter is specified, only explainable SQL statements, where an access path of hash access has been selected, are processed. After processing the results are shown in the Easy Explain report.

The following examples show the format and use of this subparameter:
```sql
PLN=ProgramA,AccType=M
Plan=SalPgm,index=T,first=486,acctype=tablespacescan;
```
In the first example, Easy Explain only selects explain data for SQL statements having matching index scan and access path.
In the second example, only SQL statements with precompiler numbers greater than or equal to 486 and where the access path is table space scan are shown in the Easy Explain report. Because the subparameter INDEX has been set to T, all indexes created for the accessed table are shown in detail.

All is the default value.

DBRM
You can specify the DBRM subparameter for the PLAN keyword to control the DBRMs for explanation within the plan.

If DBRM is not specified, all DBRMs within the plan are explained.

If you want only a given DBRM within the plan explained, the actual DBRM name must be specified.

If you want all DBRMs with a given name pattern explained, type the name string with a wildcard (%).

The following examples show the format and use of this subparameter:
```sql
Plan=ProgramA,dbrm=module27,first=17,last=34
Plan=SalPgm,dbrm=module%,level=NoKeys
```
In the first example, Easy Explain only explains module27 DBRM in the ProgramA plan. All other DBRMs in the plan are not explained.
In the second example, Easy Explain explains all DBRMs in the plan where the DBRM names start with module. For example, module02, module56, and so on.

FIRST/LAST
You can specify the FIRST or LAST subparameters for the PLAN keyword to control the range of SQL statements for explanation within the plan.

If FIRST is not specified, a value of 1 is used. If LAST is not specified, a value of 999,999,999 is used. If FIRST has a higher value than LAST, both subparameters are set to FIRST.

The following examples show the format and the use of these subparameters:
```sql
PLN=ProgramA,First=1234,Last=2345,Index=y
Plan=SalPgm,index=T,first=486
plan=bigpgm,last=4000;
```
In the first example, Easy Explain only explains SQL statements where the QUERYNO (precompiler statement number) is equal to or higher than number 1234, and less than or equal to 2345.

In the second example, only SQL statements with precompiler statement numbers greater than 485 are explained and shown in the report.

In the last example, only statements with query numbers less than or equal to 4000 are included in the report.

**FORCE**

You can specify the FORCE subparameter for the PACKAGE keyword to control the explaining of statements in a package.

If FORCE is not specified, Easy Explain only explains the statements if only one package conforms to the package specification, or if the total number of SQL statements (including non-explainable) is less than 300.

The FORCE subparameter can have the following values (only the first character is validated):

**No** If more than one package is referenced, the statements are only explained if the total number of SQL statements is less than 300.

**Yes** When this subparameter is specified, all statements in all packages that conform to the specification are explained.

The following example shows the format and use of this subparameter:

```
Package=ABC*.DEF*.(*),Force=Yes
```

The FORCE keyword forces all statements in all versions in any package name starting with DEF in any collection ID starting with ABC to be explained.

No is the default value.

**FORMAT**

You can specify the FORMAT subparameter for the PLAN and PACKAGE keyword to control the formatting of the SQL statements in the plan.

**Yes** For each explainable SQL statement in the plan, the actual SQL statement is printed in the top part of the report. The SQL statement is formatted so that a new line is started for SQL keywords such as SELECT, INTO, FROM, and WHERE. Subselects, however, are not indented.

**No** When this subparameter is specified, the explainable SQL statements are only formatted with respect to keywords like SELECT and UNION. Using this parameter, the SQL statement consumes minimum page space.

The following example shows the format and use of this subparameter:

```
Plan=SalPgm,index=T,format=No
```

The FORMAT keyword compresses the SQL statement as much as possible.

Yes is the default value.

**GEN**

You can specify the GEN subparameter for the PLAN and PACKAGE keywords to control the number of versions (or generations) in a package to be explained.

If GEN is not specified, the number of generations is one.

The following example shows the format and use of this subparameter:
The GEN subparameter results in explaining all SQL statements for Version 6 in addition to the previous version.

**HINT**

To modify an existing access path in a miniplan, you can specify an optimization hint ID. You must supply a SET CURRENT HINT statement, which updates the OPTHINT column in the PLAN_TABLE table with a specified hint name, for example, MYHINT. For more information, see [“Input command definitions” on page 160](#). In addition, you must specify a unique optimization hint ID number in the statement as shown in the following example:

```
SET CURRENT HINT = 'MYHINT', id = 4711
```

Where the number, in this case 4711, is located at the bottom of the top section of the Easy Explain report.

After you submit the Easy Explain statement, perform the following steps:

1. Change the actual access path information. For example, change from a hybrid join to a merge scan join by issuing the following SQL statement:

   ```
   UPDATE Plan_Table
   SET METHOD = 2
   WHERE METHOD = 4
   AND OPTHINT = 'MYHINT'
   ```

2. Activate the hint by issuing the following SQL statement:

   ```
   SET CURRENT OPTIMIZATION HINT = 'MYHINT';
   ```

3. Explain the SQL statement to see if the specified hint takes effect correctly.

By not including the three specified activities in Easy Explain directly, you avoid changing an access path. Changing an access path can have a significant influence on the performance of a miniplan. An access path change must only be applied by SQL programmers who understand the full impact of access paths on the performance of your systems.

**HOSTVAR**

You can specify the HOSTVAR subparameter for the PLAN and PACKAGE keywords to control the listing of host variable specifications if the application was precompiled under DB2 Version 2.3 or later.

If HOSTVAR is not specified, Easy Explain does not list any host variable specifications.

The HOSTVAR subparameter can have the following values (only the first character is validated):

- **No**: No host variable definitions are listed in the Easy Explain report.
- **Yes**: When this subparameter is specified, a listing of all (maximum 100) host variables used in the SQL statement are produced. This listing includes the definition type and length.

The following example shows the format and use of this subparameter:

```
Plan=SalPgm,Index=Yes,Hostvar=Yes
```

The HOSTVAR keyword results in a listing of all host variables and their application definitions. The definitions must be comparable to the corresponding column definitions, or the Optimizer might not choose the
optimum access path. The host variable report indicates the release number of the DB2 subsystem that was used for precompiling.

No is the default value.

INDEX
You can specify the INDEX subparameter for the PLAN, QMF, and QNO keywords to control the level of index information. The valid values for the INDEX subparameter are:

Yes
No
All
Cond
Tscn

Yes
When Yes is specified and if the Optimizer has selected a matching or non-matching index scan, Easy Explain shows all index information for the selected index including key column information. If the Optimizer has selected a table space scan, Easy Explain shows any available indexes for the accessed table.

Yes is used when CONDENSE or ALLindex is specified.

No
When No is specified, no index information is shown in the Easy Explain report.

Tablespace
When table space is specified, detailed index information for all indexes of the accessed table is shown in the Easy Explain report, if the chosen access path is table space scan. If the access path is an index scan, only information for the chosen index is shown in the report.

Condense
When Condense is specified, information for all indexes used by all tables in the plan or package is shown in one report at the end of the plan or package information. No index information is shown in the individual miniplan reports.

ALLindex
When ALLindex is specified, information for all indexes created for the table is shown in the Easy Explain report. The information is listed after the table details.

The following examples show the format and use of this subparameter:

```
PLN=SalPgm,index=NO
QMF=Agent907.Secret_query,Index=t
```

Yes is the default value.

LEVEL
You can specify the LEVEL subparameter for the PLAN and PACKAGE keywords to control the level of information within the current plan being explained. This subparameter has the same values as the LVL command, except XON and XOFF (only the first three characters are validated). Refer to the LEVEL command definition in “Input command definitions” on page 160.
Regardless of the current value set by the LVL statement, the LEVEL subparameter can specify the information level in effect for the plan being explained. When the next Easy Explain statement is processed, the information level specified by the LVL statement, is the current level.

The following examples show the format and use of this subparameter:

- Level=Detail
- PLN=ProgramA, level=NoKeys
- Plan=SalPgm, first=486, acctype=ts, level=summary

The global level is set to Detail. In the first example, the ProgramA plan is explained, but the report does not show key information because the LEVEL subparameter is set to NOKeys.

In the second example the global level is still Detail, but because the LEVEL subparameter is set to SUMMARY, the statements for this plan are only shown on the summary page. Only statements with table space scan as the access path and with query numbers greater than 485 are shown.

**LOCATION**

You can specify the LOCATION or LOC subparameter for the PLAN and PACKAGE keywords to determine the location where Easy Explain looks for the plan or package.

If LOCATION is not specified, Easy Explain uses the name of the last specified location server. If LOCATION had not been previously specified in Easy Explain, the home server location is used.

When a valid server location name is specified in the LOCATION subparameter, Easy Explain connects the application process to the specified application server. The location name must appear in the LOCATIONS column of the SYSIBM.LOCATIONS table, or be the name of the home server, that is, the local DB2 subsystem.

The following example shows the format and the use of this subparameter:

- Plan=SalPgm, Location=CPHWS1_DB2T

**OWNER**

You can specify the OWNER subparameter for the QNO keyword to control the authorization ID of the PLAN_TABLE for the current request.

The following examples show the format and the use of this subparameter:

- QNO=1234, Owner=OLFERT, Index=y
- QueryNo=9, Index=t, owner=USER

In the first example, Easy Explain selects the explain data for query number 1234 in the PLAN_TABLE owned by authorization ID OLFERT. You must have SELECT authorization to the table OLFERT.PLAN_TABLE to perform this function.

In the second example, OWNER is set to USER, which is a special value that is substituted with the authorization ID of the Easy Explain job submitter. This is also true in situations where the PARMS field in the RUN statement has been specified.

**PACKAGES**

You can specify the PACKAGES subparameter for the PLAN keyword to control the explaining of statements in packages within a plan.

If PACKAGES is not specified, only the newest version of the package is explained.
The PACKAGES subparameter has the following values (only the first character is validated):

- **Yes**: For each package in the plan, only statements in the earliest version are explained.
- **No**: When this subparameter is specified, no statements in the packages are explained.
- **All**: When this subparameter is specified, all statements in all versions of the packages are explained.

The following example shows the format and use of this subparameter:

```
Plan=SalPgm,packages=All
```

The PACKAGES keyword forces all versions in all packages for the SalPgm plan to be explained.

Yes is the default value.

**PLANINFO**

You can specify the PLANINFO subparameter for the PLAN keyword to control the printing of the plan and DBRM/package report after the first miniplan in a plan.

If PLANINFO is not specified, Easy Explain prints the plan and DBRM/package report if the authorization table specifies so.

The PLANINFO subparameter can have the following values (only the first character is validated):

- **Yes**: The plan and DBRM/package report are printed, if specified in the authorization table.
- **Mini**: When this subparameter is specified, a short report showing the plan parameters and characteristics is produced.
- **No**: When this subparameter is specified, no report is printed.

The following example shows the format and use of this subparameter:

```
Plan=SalPgm,Index=Yes,PlanInfo=No
```

Printing of the plan and DBRM or package report is suppressed.

Yes is the default value.

**TABLE**

You can specify the TABLE subparameter for the PLAN keyword to control the statements for explanation within the plan.

If TABLE is not specified, all statements within the plan are explained.

If you want only statements accessing a given table explained, the actual table name must be specified.

If you want all statements accessing tables with a given name pattern explained, type a name string that ends in a wildcard. You can end the name string with an asterisk (*) or a percent sign (%).

The following examples show the format and use of the TABLE subparameter:

```
Plan=ProgramA,table=table56
Plan=MyApple,table=sysibm.systab%,last=1234
Plan=SalPgm,table=USER.table12,level=NoKeys
```
In the first example, Easy Explain only explains the statements accessing table userid.TABLE56, where userid is the SQLID of the current user. All other statements in the plan are not explained.

In the second example, Easy Explain explains all statements with statement numbers less than or equal to 1234, where table creator is SYSIBM and table name starts with SYSTAB, like SYSTABLES and SYSTABLESPACE.

In the third example, Easy Explain only explains the statements accessing table TABLE12, where the creator is current SQLID.

**TOPA**

You can specify the TOPA subparameter for the PLAN and PACKAGE keywords to control if all the SQL statements of the plan or package are extracted and later sent to SQL PA.

If TOPA is not specified, Easy Explain does not extract all SQL statements.

The TOPA subparameter can have the following values (only the first character is validated):

- **Yes**
  
  All explainable SQL statements that cost more than 10,000 Service Units (SU) to run, or result in a union of multiple indexes, or result in a star join are extracted and later sent to SQL PA for further investigation.

- **All**
  
  When this subparameter is specified, all SQL statements of the plan or package are extracted and later sent to SQL PA.

- **No**
  
  When this subparameter is specified, no SQL statements are extracted.

The following example shows the format and use of this subparameter:

```
Plan=SalPgm,Index=Yes,Topa=All
```

All SQL statements are extracted from plan SalPgm.

Yes is the default value.

---

**Specifying language and explaining data from another user**

You can use the PARMS field to specify whether you want messages displayed in English or not. You can also use the PARMS field to extract EXPLAIN information from a PLAN_TABLE that belongs to another user.

**About this task**

When specifying the PARMS field that is part of a RUN statement, you can use the following procedure to determine the proper setting.

**Procedure**

1. If an NLS file was allocated to Easy Explain through the DD card TEXT, Easy Explain uses these NLS messages for access path and summary page messages. To overwrite this default, you can specify a parameter (NLS=NO) in the PARMS field of the DB2 RUN statement. For example:

   ```
   RUN PROGRAM(ANLEE) PLAN(ANLEE) LIB('authid.SANLOAD') PARMS('/NLS=NO')
   ```

   The result is that all messages are written in English.

2. If you want to extract EXPLAIN information from a PLAN_TABLE belonging to another user, you can specify the authorization ID of the other user in the PARMS field of the DB2 RUN statement. For example:
RUN PROGRAM(ANLEEE) PLAN(ANLEEE) LIB('authid.SANLOAD') PARMS('/USER6')

You must have SELECT authorization to USER6.PLAN_TABLE for this statement to work.

If the specified authorization ID in the PARMS field is different from the authorization ID of the job submitter, all QNO requests are selected from the PLAN_TABLE owned by the user ID specified in the PARMS field, unless the OWNER= subparameter has been specified.

Example

You can specify the two parameters in any order as none, one, or two parameters, as shown in the following examples:

RUN PROGRAM(ANLEEE) PLAN(ANLEEE) LIB('authid.SANLOAD') PARMS('/NLS=YES,USER6')
RUN PROGRAM(ANLEEE) PLAN(ANLEEE) LIB('authid.SANLOAD') PARMS('/user5,nls=no ')
RUN PROGRAM(ANLEEE) PLAN(ANLEEE) LIB('authid.SANLOAD') PARMS('/Nls=Yes')
RUN PROGRAM(ANLEEE) PLAN(ANLEEE) LIB('authid.SANLOAD')

Note: The first character in the PARMS field must be a forward slash (/). The forward slash is a PL/I requirement.

Selected authorization identification

You can specify authorization ID that Easy Explain uses to select the correct PLAN_TABLE.

To select the correct PLAN_TABLE, Easy Explain uses the authorization ID specified as the OWNER= subparameter with the QNO keyword as first choice.

If no OWNER= subparameter is specified, Easy Explain uses the authorization ID specified in the PARMS field.

If the OWNER= subparameter or the PARMS field were not specified, Easy Explain uses the authorization ID of the job submitter.

The following table shows the selected authorization ID of the PLAN_TABLE for the four main processes:

<table>
<thead>
<tr>
<th>Process (=keyword)</th>
<th>Authid of PLAN_TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL</td>
<td>Job submitter</td>
</tr>
<tr>
<td>QMF</td>
<td>Job submitter</td>
</tr>
<tr>
<td>PLN</td>
<td>Plan binder</td>
</tr>
<tr>
<td>PKG</td>
<td>Package binder</td>
</tr>
<tr>
<td>QNO</td>
<td>1. Owner parameter</td>
</tr>
<tr>
<td>QNO</td>
<td>2. PARMS field</td>
</tr>
<tr>
<td>QNO</td>
<td>3. Job submitter</td>
</tr>
</tbody>
</table>

Tip: If the authorization exit has specified a secondary authorization ID (current SQLID) or you have issued an SQLID command with a secondary authorization ID, this ID is used instead of the job submitter ID.
User authorization table

The following information describes the columns in the user authorization table.

The user authorization table has the following columns with the given contents:

**AUTHID**
The authorization ID of the user. If the authorization ID is specified as blanks, this row acts as default values for authorization IDs not found in the authorization table.

**NLS**
Use of globalization (NLS) data set for writing Easy Explain messages in the national language, if the ddname TEXT points to a data set with messages written in NLS. The default value is N.

**PLANINFO**
Creation of plan information rows in the plan history and DBRM history tables. If the authorization is given, the Easy Explain plan report contains plan information for the last four plan generations.

The default value is a Y.

**FORMATSQL**
Formatting of SQL statements, when explaining plans and packages. If this authorization is given, the SQL statement starts a new line for clauses such as FROM, WHERE, AND, and GROUP BY. The default value is Y.

**REMOTEXPL**
Explanation of remotely bound packages in a distributed environment. If authorization is given, Easy Explain shows explain information for packages found at the server location. The default value is Y.

Insert a row with specific authorizations for a specific user (authorization ID). Each row is time stamped resulting in the use of the most recent authorization row. If more than one row with a given user ID exists, the row with the most recent timestamp is used.

If Easy Explain is not able to find a row with your authorization ID, a default row with a blank user ID is used. If more than one row with a blank user ID exists, the row with the most recent timestamp is used.

The following example shows how to insert an authorization row for user AB12345:

```
insert into eee_eeeauth (authid, nls, planinfo, formatsql, remotexpl )
values ('AB12345', 'N', 'Y', 'Y', 'N' );
```

User defaults data set

Each Easy Explain user can specify their own set of run time default values.

For example, the system default value for ACCTYPE on the plan and package statement is All, but you might want the default value to be Nonmatch. Changing the default value is possible if you specify your preferences in a defaults data set. Specified values are saved in the ANLPARM user parameter file.
The system default values are specified in Chapter 5, “Easy Explain input specifications,” on page 151. The specification rules given in that chapter also apply for the user default values specifications.

The following table shows the parameters that you can set to user-defined values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>System default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCTYPE</td>
<td>Show this access type only</td>
<td>Matching</td>
<td>All</td>
</tr>
<tr>
<td>DSN8EXP</td>
<td>Use DB2 supplied stored procedure</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>INDEX</td>
<td>Index information</td>
<td>Tabscan</td>
<td>Yes</td>
</tr>
<tr>
<td>HOSTVAR</td>
<td>Show host variables</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Formatting of the SQL statement</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LEVEL</td>
<td>Level of report information</td>
<td>KEYdist</td>
<td>Detail</td>
</tr>
<tr>
<td>PLANINF</td>
<td>Level of plan information</td>
<td>Mini</td>
<td>Yes</td>
</tr>
<tr>
<td>SQLPA</td>
<td>Produce SQL Perf. Analyzer input</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SQL</td>
<td>Length of SQL statement lines</td>
<td>65</td>
<td>79</td>
</tr>
</tbody>
</table>

The user default values are specified inline in the job JCL as shown in the following example:

```
//...
//REPORT DD SYSOUT=*  
//DEFAULT DD *
acctype=Tabscan
level=AllIndex
SQL=72
//DSNTRACE DD SYSOUT=*  
//...
```

Alternative the default values are specified in a member in a PO-data set for batch jobs.

```
//...
//REPORT DD SYSOUT=*  
//DEFAULT DD DSN=hlq.kSANLPARM(ssidPARAM),DISP=SHR
//DSNTRACE DD SYSOUT=*
//...
```

**Transferring complex queries to SQL PA**

Easy Explain is able to output complex SQL queries to a data set that is used as input to SQL PA.

**Procedure**

Specify the subparameter TOPA=all to have all SQL statements in a plan or package written to the data set destined for SQL PA.

You can set TOPA=all in the User Default data set. If you specify SQLPA=No, the SQL statement is not written. If you specify the default value of SQLPA=Yes, a complex query is written to a data set specified in the DD card named //SQLPA. The format of the DD card pointing to a fixed blocked 80 bytes data set, could be as shown in the following example:
Results

The output data set with one or more complex queries separated with semicolons are sent directly to SQL PA.

When SQL statements are written to the sequential data set, a number of selected column values from the PLAN_TABLE and DSN_STATEMENT_TABLE tables are copied to a user table named authid.EEPATH. If these tables do not exist, Easy Explain creates them.
Chapter 6. DB2 SQL Performance Analyzer reporting capabilities

DB2 SQL Performance Analyzer produces three expanding levels of detail in its performance report set, from the basic Cost Summary, to the Enhanced Explain and, finally, the Detail Trace report.

Each successive report is more detailed than its predecessor, including all the previous information, plus new forecast data for the SQL statements under study. There is also a fourth summary report, SQL PA Limits, which provides a quick eye-catcher for warnings and high costs. These topics examine the contents of each level of reporting, and explain the variables that they contain, and how to relate them to DB2.

Topics:
- “DB2 SQL Performance Analyzer cost summary report”
- “DB2 SQL Performance Analyzer enhanced explain report” on page 179
- “DB2 SQL Performance Analyzer detail trace report” on page 206
- “QLIMIT (installation limits) report” on page 223
- “Predicate Analysis information” on page 226
- “Host Variables information” on page 229
- “The Easy Explain report” on page 229

DB2 SQL Performance Analyzer cost summary report

The SQL PA cost summary report provides a quick summary of costs.

Each SQL PA operation produces a minimal set of results, which contain costing information about each query examined by SQL PA. In Batch mode, these results are printed to SYSPRINT and appear on your output listing. In TSO mode, the results are written to the standard file userid.ANL.COST.LOG, always allocated by the ANLCANLI CLIST during processing. Under QMF, the interceptor program returns the same cost summary report format to the user, for all queries or only the queries that exceed the resource thresholds of the host system. The DB2 SQL Performance Analyzer stored procedure returns the same information in its output parameters.

The DB2 SQL Performance Analyzer cost summary report contains three cost descriptions to accommodate your needs:
1. The cost in resource consumption terms (CPU time, I/O, elapsed time)
2. The cost in QUNITS (Query Units)
3. The cost in monetary terms

Technical users might want CPU time (TCB and SRB) and I/O counts, to fine-tune queries by costs. Other users might prefer a single QUNITs number that sums up the total cost of each query. Or you might be interested in budgets, and prefer to see the costing in monetary terms. All three forms are always available through DB2 SQL Performance Analyzer, so the choice and preference is up to you.
A sample of the Cost Summary Report is shown in the following figure.

---

*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*

* QUERY 100000001 WILL REQUIRE 2.24139 SECONDS OF ELAPSED TIME *
* DURING WHICH 0.18020 SECONDS OF CPU TIME WILL BE CONSUMED AND *
* A TOTAL OF 6 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK *
* QUNITS 3 ESTIMATED PROCESSING COST $ 0.3395 DOLLARS *

---

Figure 48. SQL PA cost summary report

Topics:

- “Resource consumption estimates”
- “QUNITS values” on page 177
- “Charge back cost estimates” on page 178
- “Warnings and notifications” on page 178

Resource consumption estimates

The following information describes the resource consumption estimates as shown in the SQL PA cost summary report.

The first line of the report contains the QUERY number and the elapsed time. DB2 SQL Performance Analyzer assigns the QUERY number in the 100,000,000 range. SECONDS OF ELAPSED TIME indicates the DB2 SQL Performance Analyzer estimate for how long the query runs before returning data. Elapsed time, also known as response time, can be critical for online systems. Elapsed time is a computed value for INSERT, UPDATE, and DELETE statements based on the following major contributors:

- CPU processing to be done by application and DB2 address spaces
- The number of I/O requests and their type (prefetch, synchronous, and asynchronous)
- The DB2 SQL Performance Analyzer calculated estimate of I/O length (based on the disk drive speed)
- The processor usage of the attach facility, such as IMS, SPUFI, CAF, or CICS
- Known wait times for data set opens and closes, lock waits, and dynamic binds
- Any other time increments that apply to your SQL query according to DB2 SQL Performance Analyzer

DB2 SQL Performance Analyzer also includes the writing of the pages back out to disk, and logging, processes often accomplished after the query terminates. The DB2 SQL Performance Analyzer elapsed time estimate does not include network time, LAN or VTAM®, which varies depending upon the baud rate, terminal type, modem specifications, and other network characteristics. Elapsed time is presented in seconds.

The second line of the report contains CPU time as a calculated value. The value is the CPU time necessary to perform all the processing for the query, from the time it is initiated, until it terminates. This processing estimate includes, at a minimum, the CPU consumed by the attach facility (unless NONE is selected for the CONNECT parameter), all the I/O processing, including driving the sequential and list prefetches, synchronous reads, asynchronous writes, logging (if any), the stage 1 and 2 predicate processing by DB2, fetching of the rows and columns,
locking, RID list manipulation, sorting, get page calls, binding, plus many other facets of the process. CPU time is based upon privately benchmarked path lengths (instruction counts). The path lengths are derived for the many internal processes of DB2, which are then converted into CPU time using the processing power of the target host system.

Conceptually, the number of instructions for a particular operation remains constant, but they might run faster or slower on different machines because these machines process instructions at different speeds. When divided by the number of ENGINES, another parameter, and adjusted for LPARENG, the portion of the physical machine dedicated to DB2 (to account for logical partitioning), DB2 SQL Performance Analyzer can arrive at the speed of a single CPU engine. DB2 SQL Performance Analyzer is also equipped with internal mechanisms to derive the CPU speed from the MVS Service Units per Second constant. The values you assign to these parameters form the basis for the CPU time calculations.

The third line of the report contains the I/O Estimate. Only the physical I/O counts are included here, not the logical I/O. The physical I/O requirements can vary, depending on which pages are found in the buffer pool, and which must be physically read in from disk. You can control the percentage of pages found in the buffer pool using the BUFFHIT parameter.

To understand physical and logical I/O, assume that you have a query that scans 64 pages and returns 25 rows every time you run it. Unless you change the data in the table, or your selection criteria, it always scans 64 pages, and retrieve 25 rows. The 64 pages must be read physically from disk, and brought into the buffer pool, where DB2 can logically scan them (using getpage). Those 64 pages are logical I/O requests. DB2 must scan all those pages to produce your 25 rows. Now, assume that your table is being read by other users. Those other users might have left some pages in the buffer pool the last time they processed. You might no longer need to bring 64 pages into the buffer pool to satisfy your query. Maybe this time you only need 32, or next time, 12, or 40. The logical I/O requirements of your query have not changed (it is still 64 pages), but the physical I/O requirements can vary, depending on which pages are found in the buffer pool, and which must be physically read in from disk.

DB2 counts I/O in a rather peculiar way. A synchronous read I/O is a request for a single page, and it counts as one I/O. A sequential or list prefetch I/O might be for 32 pages, or 16, but it also counts as one I/O. An I/O is defined by the MVS Start I/O instruction (SIO), and it could be issued for a single 4 K page, or for many pages, chained together as a single request. The DB2 SQL Performance Analyzer I/O estimates (and estimates of DB2) do not reflect the number of pages processed. The I/O estimates reflect the number of Start I/O commands issued for physical disk pages. DB2 SQL Performance Analyzer I/O estimates include both reads and writes for tables and indexes, and work files and the DB2 log. For page counts, see the getpage statistic.

**QUNITS values**

The Query Units, or QUNITS, is a specially derived cost factor developed for users who were not familiar with resource consumption variables. The QUNITS value provides a quick assessment of the relative cost of query statements.

The QUNITS represent the raw processing requirements of each query, influenced by its I/O requirements and its instruction execution. Processing is often the most expensive resource in the system.
QUNITS are based on CPU processing and the CPU time associated with all I/O processing. The higher the QUNITS value, the more processing is required. Lower values mean quicker queries, and less resource consumption.

When comparing two queries, one query might be more CPU intensive, the other more I/O bound, while elapsed time can remain relatively constant. Using the QUNITS value can simplify the choice between which query to use.

As DB2 moves closer to a memory resident database system, the better it is to base your query optimization choices on the QUNITS value.

**Charge back cost estimates**

DB2 SQL Performance Analyzer parameter values that describe the cost of various system resources for charge back purposes can be defined.

Parameters are used to put a price on CPU time (in hours), elapsed time (terminal connect, also in hours) and I/O counts (charged per 1000). If the parameters are defined, your DB2 SQL Performance Analyzer cost estimate includes a monetary value. DB2 SQL Performance Analyzer can compute the financial cost of any query, based on the resources consumed and the assigned cost of using those resources.

A cost estimate for each query, in the national currency, can quickly build charge back cost awareness. Running a query that costs $1.22 versus a query costing $822.00 is an easy decision.

**Warnings and notifications**

The DB2 SQL Performance Analyzer cost summary report is the warning system. The warnings and notifications as they are shown on the cost summary report alert you when your query exceeds one or more of the limits set for normal resource use.

The installation team chooses and sets high water marks for CPU, I/O, elapsed time, overall monetary costs, and QUNITS. When one of these values is exceeded by the SQL PA cost estimate, DB2 SQL Performance Analyzer issues a warning message. The messages are shown in the following figure.

```
ANL5025W *** WARNING:
ESTIMATE OF 10.88977 EXCEEDS "CPU TIME" LIMIT OF 10 CPU SECONDS!
ANL5028W *** WARNING:
ESTIMATE OF 4105 EXCEEDS "I/O CALL" LIMIT OF 1000 I/O CALLS!
ANL5027W *** WARNING:
ESTIMATE OF 822.70 EXCEEDS "MONETARY" LIMIT OF 150.00 DOLLARS!
ANL5029W *** WARNING:
ESTIMATE OF 205.2498 EXCEEDS "ELAPSED TIME" LIMIT OF 120 SECONDS!
ANL5026W *** WARNING:
ESTIMATE OF 1080 EXCEEDS "SERVICE UNIT" LIMIT OF 500 QUNITS!
```

*Figure 49. SQL PA cost summary report warning messages*

These messages call your attention to the query, which you can modify to conform to acceptable standards. Set the limits high enough so that they are not easily attained by the average query. Keep values high enough that exception notifications are rarely issued.
For the enhanced explain report, DB2 SQL Performance Analyzer takes the output from DB2 Explain in the plan table and expands that information with the sizing statistics obtained from the catalog for the DB2 objects used by the query.

In this report, DB2 SQL Performance Analyzer introduces a more sentence-like structure to the Explain data, saving you the effort of trying to translate the plan table variables.

DB2 SQL Performance Analyzer reports the activities by query number, query block number, plan number, and multiple index operational sequence number, just as they are represented in the plan table by DB2.

The DB2 SQL Performance Analyzer Enhanced Explain Report is written to the file specified by the ANLREP DD card for batch, or the file named by the user as the Explain Report on the DB2 SQL Performance Analyzer Define Report Data Sets Panel under TSO.

This report is not available under the QMF intercept program because it would be too time-consuming to produce. However, long-running QMF queries can always be reexamined under DB2 SQL Performance Analyzer in either TSO or batch mode.

The enhanced explain report includes a recap of the parameters currently in effect, unless DSPPARM is set to NO. These parameter values describe the configuration and assumptions of this DB2 SQL Performance Analyzer run, and are critical factors in the DB2 SQL Performance Analyzer cost assessment and performance forecast.

An example of the parameter portion of the report is shown in the following figures.
BUFFHIT 99.59 0% OF PAGES ARE FOUND IN BUFFER POOL

CONNECT NONE ATTACH OVERHEAD (CAF|DRDA|CICS|RRSAF|NONE, ETC.)

REFRESH ANY CONSIDER VB MQTS FOR ACCESS PATHS (YES | ANY | NO)

VIADRDA +OFF+ ROUTE THIS SQL TO A REMOTE DB2 LOCATION

RETCODE NO LET RETCODE REFLECT MESSAGING LEVELS (YES|NO)

DELIMIT +OFF+ ONLY COBOL QUOTESQL OPTION REQUIRES (QUOTE | +OFF+)

DSN8EXP NO EXPLAIN USING DB2 SAMPLE STORED PROCEDURE (YES|NO)

NLSCODE ON MIXED/DBCS SUPPORT VALUES (KOR|JPN|CHT|CHS|ON|+OFF+)

KEEPPLAN NO RETAIN THE CONTENTS OF PERSONAL PLAN TABLE (YES|NO)

PROCESS +OFF+ NOSEQ IS FOR DBRMS WHICH HAVE STMT#S OUT OF SEQUENCE

NUMBERS YES YES | NO IGNORE LAST 8 CHARS FROM ANLIN SEQ FILE

Figure 50. Parameter recap for enhanced explain report
COMMENT >>>>>>> REPORTING PREFERENCES <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
COMMENT **********************************************************************

REPORTS DET CHOOSE LEVEL: LOG, REP, DET

SHOWALT YES SHOW ALTERNATE ACCESS PATHS NOT CHOSEN (YES|NO)

PRECISE ALL EXTRA COST OPTIMIZATION FOR PRECISION (YES|NO|ALL)

ADVISOR ALL WARNINGS/ALERTS + RECS/NOTES=YES, GUIDES/GOOD=ALL

OBJECTS YES PTF3104 SHOW ALL TABLES AND INDEXES ACCESSED BY SQL

QLIMSQL YES SHOW SQL THAT FAILED THE EXPLAIN PREPARE (QLIMITS)

MESSAGE YES IF NO, THEN NO ADVISOR MESSAGES PRINT (JUST MSGIDS)

NOTSCAN NO IF YES, EVERY TABLESPACE SCAN WILL GET MESSAGE 5042W

NOSTATS YES ISSUE WARNING IF NO STATS FOR OBJECTS, MESSAGE 3026W

ALLPART YES ISSUE WARNING ANL5044W IF ALL PARTITIONS ARE TSCANED

NONONIX NO IF YES, EVERY NONMATCH INDEX WILL GET MESSAGE 5043W

TURNOFF 9999 USED TO 'TURNOFF' MESSAGE IDS FROM THE ADVISOR LIST

DSPVARS NO YES DISPLAY HOST VARS, NO DISPLAY PARAMETER MARKERS

DSPPARM YES YES | NO DISPLAY PARAMETERS USED ON TOP OF REPORTS

COMMENT **********************************************************************

COMMENT >>>>>>> SET LIMITS FOR REPORTING MESSAGES WHEN LIMIT IS EXCEEDED

COMMENT **********************************************************************

JOINTAB 10 LIMIT TO NUMBER OF JOIN TABLES BEFORE MESSAGE 7017I

INLISTS 10 LIMIT NUMBER ELEMENTS IN(LIST) BEFORE MESSAGE 5007W

IXUPDAT 5 LIMIT NUMBER INDEXES ON UPDATE BEFORE MESSAGE 6001I

TSCANPG 50000 NUMBER OF PAGES READ BY TSCAN BEFORE MESSAGE 6066I

ISCANPG 1000 NUM LEAF PAGES IN MATCH IX SCAN BEFORE MESSAGE 6067I

NONIXPG 5000 NUM LEAF PAGES IN NONMATCH SCAN BEFORE MESSAGE 6068I

MATICOLS 0.50 LIMIT PERCENT COLUMNS MATCHED IN SCAN, MESSAGE 6061I

PREDPCT 0.15 LIMIT PCT ROWS FILTERED BY PREDICATES, MESSAGE 6025I

COMMENT **********************************************************************

Figure 51. Parameter recap for enhanced explain report (continued)
COMMENT >>>>>> PERCENTAGE OF PAGES <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
COMMENT ***************************************************************
COMMENT >>>>>> DEVICE TYPE FOUND IN THE USERS GUIDE APPENDIX C <<<<<<<<
COMMENT ***************************************************************

STORAGE ESS800 THE DASD FARM IS IBM 'SHARK' ESS SOLID STATE DRIVES
(ANLPARM)

DEGREES 1

DBRMKEY DBRMNAME
EXPLAIN NEW
EEECALL SPA
QUALIFY PROD

USEPLAN PROD
QUERYNO 1
STOPQNO 999999999

VERSION N/A will be discovered by code
(ANLCNTL)

COMMENT ***************************************************************

COMMENT ***************************************************************
COMMENT >>>>>> PARAMETERS DISCOVERED DURING INSTALL <<<<<<<<<<<<<<<<<<<
COMMENT ***************************************************************

ENGINES 10 ENGINES

BUFFERS 00020000 SIZE OF 4K APPLICATION BUFFER POOL
BUFF08K 00000500 8K BUFFER POOLS FOR APPL
BUFF16K 00000250 16K BUFFER POOLS FOR APPL
BUFF32K 00000100 SIZE OF 32K APPLICATION BUFFER POOL
DYNAMIC YES CACHING OF DYNAMIC SQL STATEMENTS (YES|NO)

COMMENT ***************************************************************

Figure 52. Parameter recap for enhanced explain report (continued)
COMMENT >>>>>> SET THE RUN TIME ENVIRONMENT <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

DBTRACE OFF DIAGNOSTICS TRACE PARM: OFF | ON | ALL | DMP

SETPLAN YES AUTHORIZE USER TO SPECIFY HLQ FOR PLAN_TABLE

ANLKEYS 00000 PARSER=01000 CLASS1=10000 NOVER=02000 NOPT=20000

COMMENT ************************************************************

COMMENT >>>>>> HARDWARE/SOFTWARE ENVIRONMENT <<<<<<<<<<<<<<<<<<<<<<<<<<<

LPARENG 01.000 PORTION OF THE BOX DEDICATED TO DB2

ESASORT YES HARDWARE ASSISTED SORT IS ACTIVE

ESACOMP YES HARDWARE ASSISTED DATA COMPRESSION IS ACTIVE

COMMENT **************************************************************************

COMMENT >>>>>> ADJUST THE CPU OVERHEAD IN A DATA SHARING ENV <<<<<<<<<<<<<<<

DSGROUP 0 # OF MEMBERS IN THE DATA SHARING GROUP, 0-32

DATASHR 0 AVG % WORKLOAD PARTICIPATES IN DATA SHARING, 0-100

COMMENT **************************************************************************

COMMENT >>>>>> SET COST LIMITS USED FOR WARNING MESSAGES <<<<<<<<<<<

CPUTIME 00010 10 SECS OF CPU TIME

COSTING 0000015 $15.00 QUERY COST (MONETARY TERMS)

ELAPSED 00060 1 MINUTE ELAPSED TIME

COSTQUN 00000200 200 QUERY SERVICE QUNITS

IOCALLS 00001000 1 000 I/O CALLS DISK I/O PHYSICAL

Figure 53. Parameter recap for enhanced explain report (continued)

Topics:

- “Table space scans” on page 184
- “Cluster matching index scans” on page 186
- “Cluster nonmatching index scans” on page 188
- “Random matching index scans” on page 189
- “Random nonmatching index scans” on page 191
- “Multiple index scans” on page 192
- “Joining tables” on page 194
- “Subquery processes” on page 198
- “Sorting results” on page 201
- “Insert, Update, Delete, Select, and Refresh” on page 202
Table space scans

The table space scan information includes the type of table space (simple, segmented, or partitioned), the number of tables in the space, and the fully qualified table name. The size of the table in pages, its record length, number of records, and columns are also provided. Other pertinent information, such as lock size and table space lock mode, close option, segment size, and percent compressed, completes the description of the table space.

This descriptive data is presented for each table accessed, detailing the size, scope, and contents of the tables being accessed by the query. The DB2 SQL Performance Analyzer Cost Summary Report is presented along with the Explain information so all the information is found in a single place. This run contains the parameter SHOWALT YES, so alternative indexes available to access this table are also presented.

When the optimizer for DB2 chooses a table space scan as the access path, it is ignoring the use of indexes, if any exist on the table, as a means of accessing the data. There are many reasons why you cannot choose an index, and the optimizer has made a valid choice in access path selection.

Table space scans are valuable when columns do not have predicates that participate in these indexes. There is no benefit in using the indexes because they would not eliminate any rows from the search.

Occasionally, the indexed columns might be involved with predicates that are stage 2, or contain subtle incompatibilities, such as being compared to columns or data of the wrong length or type.

DB2 SQL Performance Analyzer explains the table space scan in the report segment shown in the following example. Notice that the query states “FROM L1000,” but SQL PA adds the high-level qualifier TDT690 to the table name.

```
EXPLAIN PLAN SET QUERYNO = 100000001 FOR
SELECT * FROM L1000
```

```
100000001 1 1 0
-----

ANL7003I *** GUIDELINE:
Close Yes was specified for the Tablespace and/or the index... if these are little used this is OK. If high volume access then consider Close No. Extremely relevant pages can be "page fixed" in memory by highly referencing, using Hiperpools (Castout Y/N), putting into a dedicated buffer pool large enough to hold all pages, deploying data sharing with Group Buffer Pool Cache All option, etc. each with associated costs. Close No also increases chances that DBD will remain in EDM Pool for next execution.

ANL7006I ***
This statement contains a select of all columns in the table. Typically a select of all columns results in significantly increased processing time to fetch and process each column of each row. Use select all only when you want to select all columns in a view definition. Improve performance by specifying the name of each column for the SQL SELECT statement.
```
ANL5000W *** WARNING:
This SQL statement contains no predicates and uses no Built in Functions, so DB2 is selecting the Tablespace Scan access method. In general, this means poor performance, unless the table is very small: verify the accuracy of catalog statistics (NPAGES, NACTIVE, CARD), and re-evaluate whether this access path is appropriate.

TABLESPACE SCAN
------------
CREATOR: TDT690
TABNAME: L1000

TABLESPACE SCAN WAS CHOSEN ON A 4K SEGMENTED TSPACE WITH: 3 TABLES
VERSION: 0 TABLE CONTAINS A TOTAL OF: 31680 4K PAGES
TABLE ROWS: 100000 COLUMNS: 50 REC LENGTH: 1008 BYTES
DSSIZE: 0 GB NUMBER OF MQTS: 0 LOG: Y AVG ROW LENGTH: 1006 BYTES
TYPE: T LOCK SIZE: A TS LOCK MODE: IS LOCK PART: N CLOSE TABLE: Y
PAGES WITH ROWS: 99% PCT COMPRESSED: 0% MAX ROWS: 255 BPOOL: BP11
ENCODE: E CCSIDS ARE SBCS: 833 DBCS: 834 MIXED: 933 VOLATILE: N

SEQUENTIAL PREFETCH WILL BE EMPLOYED TO ACCESS THIS TABLE

ALTERNATIVE INDEX
+++++++++++++++++
CREATOR: TDT690
IX NAME: L1000R1N

VERS: 0 KEY LEN: 6 PADDED: - C-ED: N C-ING: N CLURATIO: 11.6627
FULLKEY CARD: 5954 FIRSTKEY CARD: 5954
TYPE: 2 NLEAF PAGES: 155 NLEVELS: 2 UNIQUE: D KEY COLS: 1

KEY ORDER COLCARD COLUMN NAME
--------------------------------
  1 A 5954 RIKEY1

ALTERNATIVE INDEX
+++++++++++++++++
CREATOR: TDT690
IX NAME: L1000R2N

VERS: 0 KEY LEN: 6 PADDED: - C-ED: N C-ING: N CLURATIO: 70.0010
FULLKEY CARD: 100000 FIRSTKEY CARD: 100000
TYPE: 2 NLEAF PAGES: 359 NLEVELS: 3 UNIQUE: U KEY COLS: 1

KEY ORDER COLCARD COLUMN NAME
--------------------------------
  1 A 100000 RIKEY2

* QUERY 10000001 WILL REQUIRE 95.28095 SECONDS OF ELAPSED TIME *
* DURING WHICH 26.08726 SECONDS OF CPU TIME WILL BE CONSUMED AND *
* A TOTAL OF 6967 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK *
* QUNITS 199 ESTIMATED PROCESSING COST $ 14.0652 DOLLARS *

ANL5025W *** WARNING: ESTIMATE OF 95.281 EXCEEDS "ELAPSED TIME" LIMIT OF 60 SECONDS!
Cluster matching index scans

The cluster matching index scan is one of the preferred access paths chosen by the optimizer.

The optimizer uses a clustered index because the data and index are in the same general order at least 80% of the time. DB2 matches one or more columns against that index, qualifying the predicates at the index level. This matching often results in a RID (row identification) list that, when applied against the data table, is shorter and contains fewer pages to be read because the rows follow the same order as the index.

For example, if the predicate being evaluated was WHERE COL1 BETWEEN 10 AND 100 and a clustered index existed on COL1. Most of the rows that contained COL1 values between 10 - 100 would be contiguously placed in the data table, following each other in order, such as 10, 11, 12, 20, 21, 30... 100. If 5 rows fit per page, and 100 rows were returned, then only 20 pages would need to be read to satisfy the query.

Suppose, however, that the index was not clustered. Then there would be no guarantee that the rows would follow each other, or even be on the same page. In fact, it might take up to 100 pages to read in the 100 rows, if the distribution were random.

A cluster matching index scan reads a portion of the index leaf tree, and a portion of the data table pages to satisfy the query. It is one of the best alternatives for table access. If a significant number of data pages are read, sequential prefetch is wanted to retrieve them, and the same is true for index leaf pages.

Detailed information is provided about the clustered index wanted, including its cluster ratio, type, and cardinality. The number of columns which are matched by DB2 for predicates on that index is also included. These are always Stage 1 predicates. The index unique rule is presented as DUPS_OK, UNIQUE or PRIMARY (which also means unique), and CONST-U (unique constraint), NOTNULL (for unique where not null), and PARENT for the non-primary parent keys.

The size of the index, from the number of levels to the number of leaf pages is described, along with the index close option, first and full key cardinality (the number of unique values in the first column of the index, and in all the columns of index), along with other pertinent index descriptive data, including the index key composition, column cardinality and nonuniform distributions. SHOWALT YES brings the additional information about the alternative indexes that the optimizer could have selected, but did not.

The table on which the index is operating is also described, providing information like the table space scan information. Special notes, such as the IN predicate being used as the matching predicate, provide you with further information about the current access. If the clustered index also has the CLUSTERING flag set (only one does), it is marked as the Insert controlling index, containing the order in which all new rows must be inserted.

The enhanced explain report information for cluster index matching scans is shown in the following example:

```
EXPLAIN PLAN SET QUERYNO = 100000002 FOR
SELECT CIKEY, C2, NIKEY, C4, RIKEY1, C6 FROM TDT690.L1000
  WHERE CIKEY IN (50087, 50088, 50089, 50090, 70090) 
```
PROCESS ->

THIS INDEXED ACCESS USES THE "IN (LIST)" PREDICATE, WHICH IS NOT A CANDIDATE FOR MULTIPLE INDEX ACCESS.

+------------------------------------------------------------------+
|ANL7002I *** GUIDELINE:                                          |
|This plan step has not selected any Sequential|List Prefetch I/O. If the SQL processes just a few rows that is OK, but if many rows are involved, you can help promote Sequential Detection by both accessing the data in sequential order (presort?) and by binding with Release (Deallocate) to avoid resetting counters at Commit. |
+------------------------------------------------------------------+

+------------------------------------------------------------------+
|ANL7003I *** GUIDELINE:                                          |
|Close Yes was specified for the Tablespace and/or the index... if these are little used this is OK. If high volume access then consider Close No. Extremely relevant pages can be "page fixed" in memory by highly referencing, using Hiperpools (Castout Y/N), putting into a dedicated buffer pool large enough to hold all pages, deploying data sharing with Group Buffer Pool Cache All option, etc. each with associated costs. Close No also increases chances that DBD will remain in EDM Pool for next execution. |
+------------------------------------------------------------------+

CLUSTER MATCH IX SCAN
---------------------
IX CREATOR: TDT690
INDEX NAME: L1000CIN

VERS: 0 KEY LEN: 6 PADDED: - C-ED: Y C-ING: Y CLURATIO: 99.9995
FULLKEY CARD: 99340 FIRSTKEY CARD: 99340
TYPE: 2 NLEAF PAGES: 412 NLEVELS: 3 UNIQUE: D DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10

KEY COLUMN NAME ORDER TYPE DIST LEN NULL COLCARD DIST#
------------------------------------------------------------------------
1 CIKEY A DECIMAL N 9 Y 99340 10

ALTERNATIVE INDEX
+++++++++++++++++
CREATOR: TDT690
IX NAME: L1000R1N

VERS: 0 KEY LEN: 6 PADDED: - C-ED: N C-ING: N CLUSRATIO: 11.6627
FULLKEY CARD: 5954 FIRSTKEY CARD: 5954
TYPE: 2 NLEAF PAGES: 155 NLEVELS: 2 UNIQUE: D KEY COLS: 1

KEY ORDER COLCARD COLUMN NAME
---------------------------------
1 A 5954 RIKEY1

ALTERNATIVE INDEX
+++++++++++++++++
CREATOR: TDT690
IX NAME: L1000R2N

VERS: 0 KEY LEN: 6 PADDED: - C-ED: N C-ING: N CLUSRATIO: 70.0010
FULLKEY CARD: 100000 FIRSTKEY CARD: 100000
TYPE: 2 NLEAF PAGES: 359 NLEVELS: 3 UNIQUE: U KEY COLS: 1

KEY ORDER COLCARD COLUMN NAME
---------------------------------
Cluster nonmatching index scans

The cluster nonmatching index scan also uses a clustered index, but unlike the cluster matching index scan, the columns are not matched against its specific values.

Rather, the entire index leaf tree structure is read, and the predicate is applied against the column values to arrive at a conclusion. For example, if the predicate were WHERE COL1 NOT LIKE 'ABC', all the values of COL1 must be processed from the index to determine which ones are not like 'ABC'. The nonmatching index scan is used when columns are involved in a summarization built-in function, such as AVG or SUM. The nonmatching index scan is used in a stage 2 predicate process, as shown in the following example.

Notice that in this example no columns are matched against the index. Also note that in this query, the column involved in the Stage 2 arithmetic expression predicate is found in the index, and no others are selected. Only the index is accessed, as noted by SQL PA. Cluster nonmatching index scans make heavy use of sequential prefetch for both index and data pages.

This example also provides a look at the evaluation of the query by SQL Advisor, with informative notes for the user.
EXPLAIN PLAN SET QUERYNO = 100000003 FOR
SELECT CIKEY FROM L1000
WHERE CIKEY - 200.50 < 2000;

QUERYNO: 100000003 QBLOCKNO: 1 PLANNO: 1 MIXSEQ: 0
PROCESS ->

CLUSTER NONMATCH IX SCAN
-----------------------
IX CREATOR: TDT690
INDEX NAME: L1000CIN
VERS: 0 KEY LEN: 6 PADDED: - C-ED: Y C-ING: Y CLUSRATIO: 99.995
FULLKEY CARD: 99340 FIRSTKEY CARD: 99340
TYPE: 2 NLEAF PAGES: 412 NLEVELS: 3 UNIQUE: D KEY COLS: 1
NONE OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10

KEY COLUMN NAME ORDER TYPE DIST LEN NULL COLCARD DIST#
------------------------------------------------------------------------
1 CIKEY A DECIMAL N 9 Y 99340 10

+------------------------------------------------------------------+
| ANL6017I *** NOTE: |
| This query is currently performing a Nonmatching Index Scan, which |
| causes it to read every index Leaf page to process the predicates. |
| Reading the entire leaf tree is time-consuming vs. matching a part |
| of the index. Consider better predicates that allow you to match. |
+------------------------------------------------------------------+

| ANL6020I *** NOTE: |
| This index is presently designated as allowing "duplicate values". |
| Make sure this is a nonunique index that does not fall into one of |
| the following "unique" categories: explicitly unique, primary key, |
| non-primary RI parent key, unique where not null, unique column |
| constraint. Unique indexes have definite advantages wherever they |
| can be declared by the user, so be sure it contains duplicates. |
+------------------------------------------------------------------+

THIS IS AN "INDEX ONLY" ACCESS: NO DATA PAGES ARE READ FROM THE TABLE.

| ANL6025I *** NOTE: |
| The Composite (final) Filter Factor indicates that more than 15% |
| of the rows will be returned: predicates not very restrictive |
+------------------------------------------------------------------+

| ANL6026I *** NOTE: |
| Arithmetic Expressions that are not exclusively using integers |
| will be Stage 2 and not matching indexable. Move to Select list |
| if possible, or do the math in your application later. |
+------------------------------------------------------------------+

SEQUENTIAL PREFETCH WILL BE EMPLOYED DURING THIS INDEX ONLY ACCESS

* QUERY 100000003 WILL REQUIRE 7.61600 SECONDS OF ELAPSED TIME *
* DURING WHICH 0.23053 SECONDS OF CPU TIME WILL BE CONSUMED AND *
* A TOTAL OF 15 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK *
* QUNITS 3 ESTIMATED PROCESSING COST $ 0.2641 DOLLARS *

Random matching index scans
The random matching index scan is like the cluster matching index scan with two major differences.
The differences include the following characteristics:

- The rows in the data table are not clustered together in the order of the index, but rather scattered randomly throughout the pages of the table.

- The list prefetch is used to retrieve the data pages. The RIDs are sorted into the proper order of appearance (table page order), so that the list prefetch I/O is sequential, if not contiguous, in nature. An example of the information for a random matching index scan is shown in the following information.

DB2 SQL Performance Analyzer provides the same basic information for a random index as it does for a clustered index and, if there is table access, the form and structure of the table is also shown.

```sql
EXPLAIN PLAN SET QUERYNO = 100000004 FOR
SELECT * FROM L1000
WHERE RIKEY2 = 38;
```

<table>
<thead>
<tr>
<th>QUERYNO: 100000004</th>
<th>QBLOCKNO: 1</th>
<th>PLANNO: 1</th>
<th>MIXOPSEQ: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS -&gt;</td>
<td>RANDOM MATCH IX SCAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IX CREATOR: TDT690</td>
<td>INDEX NAME: L1000R2N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VERS: 0 KEY LEN: 6 PADDEN: - C-ED: N C-ING: N CLUSRATIO: 70.0010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULLKEY CARD: 100000 FIRSTKEY CARD: 10000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYPE: 2 NLEAF PAGES: 359 NLEVELS: 3 UNIQUE: U KEY COLS: 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KEY COLUMN NAME ORDER TYPE DIST LENGTH NULL COLCARD DIST#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIKEY2 A DECIMAL N 9 Y 100000 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ALTERNATIVE INDEX**

| CREATOR: TDT690 | IX NAME: L1000CIN |
| VERS: 0 KEY LEN: 6 PADDEN: - C-ED: Y C-ING: Y CLUSRATIO: 99.9995 |
| FULLKEY CARD: 99340 FIRSTKEY CARD: 99340 |
| TYPE: D NLEAF PAGES: 412 NLEVELS: 3 UNIQUE: D KEY COLS: 1 |
| KEY ORDER COLCARD COLUMN NAME |
| CIKEY A 99340 |

**ALTERNATIVE INDEX**

| CREATOR: TDT690 | IX NAME: L1000R1N |
| VERS: 0 KEY LEN: 6 PADDEN: - C-ED: N C-ING: N CLUSRATIO: 11.6627 |
| FULLKEY CARD: 5954 FIRSTKEY CARD: 5954 |
| TYPE: 2 NLEAF PAGES: 155 NLEVELS: 2 UNIQUE: D KEY COLS: 1 |
| KEY ORDER COLCARD COLUMN NAME |
| RIKEY1 A 5954 |

**MATCHING IX SCAN**

| CREATOR: TDT690 | TABNAME: L1000 |
| THIS INDEXED ACCESS IS APPLIED TO 32K SEGMENTED SPACE WITH: 3 TABLES |
| VERSION: 1 TABLE CONTAINS A TOTAL OF: 33335 4K PAGES |
Random nonmatching index scans

The random nonmatching index scan is seldom chosen as an access path by DB2. It is not efficient to read the entire index leaf tree, and one page per row retrieved (the assumption of the optimizer).

However, certain built-in functions, left unqualified by further WHERE clause predicates, are processed best using a random nonmatching index scan.

It is not uncommon to see the COUNT all rows in the table function processed as a nonmatching index scan against the smallest index on the table that counts all the RIDs that are encountered. The random nonmatching index scan displays the complete set of index statistics and includes the table characteristics, if the query requires rows from the data table. The query in the example does not.

The following example uses the COUNT(*) clause:

```
EXPLAIN PLAN SET QUERYNO = 100000005 FOR
SELECT COUNT(*) FROM L1000
```

**ANL7003I *** GUIDELINE:**

Close Yes was specified for the Tablespace and/or the index... if these are little used this is OK. If high volume access then consider Close No. Extremely relevant pages can be "page fixed" in memory by highly referencing, using Hiperpools (Castout Y/N), putting into a dedicated buffer pool large enough to hold all pages, deploying data sharing with Group Buffer Pool Cache All option, etc. each with associated costs. Close No also increases chances that DBD will remain in EDM Pool for next execution.

**--- RANDOM NONMATCH IX SCAN ---**

| IX CREATOR: TDT690 |
| INDEX NAME: L1000R1N |
| VERS: 0 | KEY LEN: 6 PADD: - C-ED: N C-ING: N CLUSRATIO: 11.6627 |
| FULLKEY CARD: 5954 | FIRSTKEY CARD: 5954 |
| TYPE: 2 | NLEAF PAGES: 155 NLEVELS: 2 UNIQUE: D KEY COLS: 1 |
| NONE OF 1 COLUMNS ARE MATCHED | CLOSE: Y LOCK MODE: IS BPOOL: BP10 |
| KEY COLUMN NAME | ORDER | TYPE | DIST | LEN | NULL | COLCARD | DIST# |
| 1 RIKEY1 | A | DECIMAL | N | 9 | Y | 5954 | 10 |
**Multiple index scans**

Sometimes DB2 chooses to process a query using multiple index scans, building a composite RID list by ANDing and ORing RIDs which match each predicate, until one final RID list is created. The list is then used to access the data table pages using list prefetch.

For multiple index scans, several points are worth noting. The MIXSEQ (multiple index operational sequence) number identifies each step within a PLANNO (plan number) that is taken during the multiple index access. Each index is a matching scan, done as index only. The same index can appear several times in the sequence. If the predicates are linked by AND, the intersection of the two RID lists is used because a row satisfying the predicate clause must exist in both lists. If the predicates are joined by OR, the union of the two RID lists is used because a row satisfying one of the predicates could appear either on RID list A or RID list B. SQL PA explains each of the steps in this process, including the final data access using List Prefetch.
The following example shows a sample report for a multiple index scan. Three different indexes were used for the Multiple Index access query in the sample. In step 1 the CIN index was accessed and, in step 2, the R1N index. These are intersected in step 3 (MI) to create a composite list made up of qualifying RIDs. Then the R2N index is accessed in step 4, and its RIDs are unioned (MU) with the composite RID list in step 5. Finally, data was accessed using list prefetch. DB2 Explain identifies this action as step 0 (MIXSEQ = 0), but it is performed last, not first.

EXPLAIN PLAN SET QUERYNO = 100000006 FOR
SELECT * FROM TDT690.L1000
  WHERE CIKEY BETWEEN 1001 AND 1100 AND RIKEY1 < 10000 OR
   RIKEY2 BETWEEN 52 AND 1500

QUERYNO: 100000006 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 0
PROCESS ->
A MULTIPLE INDEX OPERATION HAS BEEN REQUESTED TO ACCESS THIS TABLE:

QUERYNO: 100000006 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 1
PROCESS ->
CLUSTER MATCH IX SCAN
---------------------
IX CREATOR: TDT690
INDEX NAME: L1000CIN

VERS: 0 KEY LEN: 6 PADDED: - C-ED: Y C-ING: Y CLUSRATIO: 99.9995
FULLKEY CARD: 99340 FIRSTKEY CARD: 99340
TYPE: 2 NLEAF PAGES: 412 NLEVELS: 3 UNIQUE: D DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10

<table>
<thead>
<tr>
<th>KEY</th>
<th>COLUMN NAME</th>
<th>ORDER</th>
<th>TYPE</th>
<th>DIST</th>
<th>LEN</th>
<th>NULL</th>
<th>COLCARD</th>
<th>DIST#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CIKEY</td>
<td>A DECIMAL</td>
<td>N</td>
<td>N</td>
<td>9</td>
<td>Y</td>
<td>99340</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

THIS IS THE CLUSTERING ("INSERT & LOAD ORDER") INDEX FOR THIS TABLE
THIS IS AN "INDEX ONLY" ACCESS: NO DATA PAGES ARE READ FROM THE TABLE

QUERYNO: 100000006 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 2
PROCESS ->
RANDOM MATCH IX SCAN
----------------------
IX CREATOR: TDT690
INDEX NAME: L1000R1N

VERS: 0 KEY LEN: 6 PADDED: - C-ED: N C-ING: N CLUSRATIO: 11.6627
FULLKEY CARD: 5954 FIRSTKEY CARD: 5954
TYPE: 2 NLEAF PAGES: 195 NLEVELS: 2 UNIQUE: D DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10

<table>
<thead>
<tr>
<th>KEY</th>
<th>COLUMN NAME</th>
<th>ORDER</th>
<th>TYPE</th>
<th>DIST</th>
<th>LEN</th>
<th>NULL</th>
<th>COLCARD</th>
<th>DIST#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 RIKEY1</td>
<td>A DECIMAL</td>
<td>N</td>
<td>N</td>
<td>9</td>
<td>Y</td>
<td>5954</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
Joining tables

DB2 SQL Performance Analyzer provides information about multi-table accesses, or joins. Information that SQL PA can ascertain about the join sequence is presented in the report. The report included information about the inner and outer table selection, and the type of join processing.
The merge scan join method is also explained by DB2 SQL Performance Analyzer.

The optimizer does much analysis before deciding the best join method, the processing order of the tables, and the inner and outer tables in the joining process. SQL PA describes the chosen approach, pointing out the access methods used to retrieve rows from each table in the process, and the inner and outer table order and pairing.

In the following example, the nested loop join method is shown:

```
EXPLAIN PLAN SET QUERYNO = 100000007 FOR
SELECT A.CIKEY, A.C2, A.NIKEY, A.C4, A.RIKEY1,
     B.C6,B.C7,B.C8,B.C9,B.RIKEY2
FROM TDT690.S100 A, TDT690.L1000 B
WHERE (A.RIKEY2 BETWEEN 20 AND 270)
     AND (A.RIKEY2 = B.RIKEY2)
```

The merge scan join method is also explained by DB2 SQL Performance Analyzer.
CLUSTER MATCH IX SCAN
---------------------
IX CREATOR: TDT690
INDEX NAME: S100RI2N

VERS: 0  KEY LEN: 6  PADDED: -  C-ED: N  C-ING: N  CLUSRATIO: 85.1000
FULLKEY CARD: 1000  FIRSTKEY CARD: 1000
TYPE: 2  NLEAF PAGES: 5  NLEVELS: 2  UNIQUE: D  DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED  CLOSE: Y  LOCK MODE: IS  BPOOL: BP10

KEY COLUMN NAME ORDER TYPE DIST LEN NULL COLCARD DIST#
------------------------------------------------------------------------
1  RIKEY2  A  DECIMAL  N  9  Y  1000  10

INDEXED ACCESS
---------------
CREATOR: TDT690
TABNAME: S100

INDEXED ACCESS IS APPLIED TO A 4K SEGMENTED TSPACE WITH: 3 TABLES
VERSION: 0  TABLE CONTAINS A TOTAL OF: 31680 4K PAGES
TABLE ROWS: 1000  COLUMNS: 10  REC LENGTH: 108 BYTES
DSSIZE: 0 GB  NUMBER OF MOTS: 0  LOG: Y  AVG ROW LENGTH: 106 BYTES
TYPE: T  LOCK SIZE: A  TS LOCK MODE: IS  LOCK PART: N  CLOSE TABLE: Y
PAGES WITH ROWS: 87%  PCT COMPRESSED: 0%  MAX ROWS: 255  BPOOL: BP11
ENCOD: E  CCSIDS ARE SBCS: 833  DBCS: 834  MIXED: 933  VOLATILE: N
THE CORRELATION NAME ASSOCIATED WITH THIS TABLE IS: A

THIS TABLE IS JOINED WITH THE PRIOR TABLE VIA THE "NESTED LOOP" METHOD.
THIS IS THE INNER TABLE IN THE NESTED LOOP JOIN.

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
* QUERY 100000007 WILL REQUIRE 103.91500 SECONDS OF ELAPSED TIME *
* DURING WHICH 68.15169 SECONDS OF CPU TIME WILL BE CONSUMED AND *
* A TOTAL OF 7 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK *
* QUNITS 3722 ESTIMATED PROCESSING COST $ 15.5035 DOLLARS *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

ANL5025W *** WARNING:
ESTIMATE OF 68.152 EXCEEDS "CPU TIME" LIMIT OF 10 CPU SECONDS!

ANL5026W *** WARNING:
ESTIMATE OF 3722 EXCEEDS "SERVICE UNIT" LIMIT OF 200 QUNITS!

ANL5027W *** WARNING:
ESTIMATE OF 15.50 EXCEEDS "MONETARY" LIMIT OF 15.00 DOLLARS!

ANL5029W *** WARNING:
ESTIMATE OF 103.915 EXCEEDS "ELAPSED TIME" LIMIT OF 60 SECONDS!

In the following example, the hybrid join method is shown:

EXPLAIN PLAN SET QUERYNO = 100000008 FOR
SELECT A.CIKEY, A.C2, A.NIKEY, A.C4, A.RIKEY1,
       B.C6, B.C7, B.C8, B.C9, B.RIKEY2
FROM TDT690.S100 A, TDT690.L1000 B
WHERE (A.CIKEY BETWEEN 505 AND 550)
       AND (A.CIKEY = B.CIKEY)

QUERYNO: 100000008  QBLOCKNO: 1  PLANNO: 1  MIXOPSEQ: 0
PROCESS ->

CLUSTER MATCH IX SCAN
---------------------
IX CREATOR: TDT690
INDEX NAME: S100RI2N
1 << 2

### Cluster Information

<table>
<thead>
<tr>
<th>Key Column Name</th>
<th>Order</th>
<th>Type</th>
<th>Dist</th>
<th>Len</th>
<th>Null</th>
<th>Colcard</th>
<th>Dist#</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIKEY</td>
<td></td>
<td>DECIMAL</td>
<td>N</td>
<td>9</td>
<td>Y</td>
<td>839</td>
<td>10</td>
</tr>
</tbody>
</table>

This is the clustering ("Insert & Load Order") index for this table.

### Indexed Access

Creator: TDT690
Table Name: S100

Indexed access is applied to a 4K segmented tspace with: 3 tables
Version: 0
Table contains a total of: 31680 4K pages
Table rows: 1000 columns: 10 rec length: 108 bytes
dssize: 0 GB number of mQts: 0 log: y avg row length: 106 bytes
Type: t
lock size: a ts lock mode: is lock part: n close table: y
Pages with rows: 87% pct compressed: 0% max rows: 255 bpool: Bp11
Encode: e
ccsids are sBCs: 833 dBCs: 834 mixed: 933 volatile: n
The correlation name associated with this table is: a

A join of 2 tables has been detected. This was the first table access.

---

<table>
<thead>
<tr>
<th>Key Column Name</th>
<th>Order</th>
<th>Type</th>
<th>Dist</th>
<th>Len</th>
<th>Null</th>
<th>Colcard</th>
<th>Dist#</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIKEY</td>
<td></td>
<td>DECIMAL</td>
<td>N</td>
<td>9</td>
<td>Y</td>
<td>99340</td>
<td>10</td>
</tr>
</tbody>
</table>

This is the clustering ("Insert & Load Order") index for this table.

### Indexed Access

Creator: TDT690
Table Name: L1000

Indexed access is applied to a 4K segmented tspace with: 3 tables
Version: 0
Table contains a total of: 31680 4K pages
Table rows: 100000 columns: 50 rec length: 1008 bytes
dssize: 0 GB number of mQts: 0 log: y avg row length: 1006 bytes
Type: t
lock size: a ts lock mode: is lock part: n close table: y
Pages with rows: 99% pct compressed: 0% max rows: 255 bpool: Bp11
Encode: e
ccsids are sBCs: 833 dBCs: 834 mixed: 933 volatile: n
The correlation name associated with this table is: b

List prefetch (skip seq.) will be employed to access this table.

This table is joined with the prior table via the "hybrid join" method.
This is the inner table in the hybrid join, accessed by its rid list.
Subquery processes

DB2 subquery processing is like join processing, except that the queries are coded as table accesses, one within another, rather than collectively from a set of tables.

Subqueries are correlated using a correlation name to associate columns from one query with tables from another query. A correlated subquery results in a repeated step wise run of the inner and outer queries, looking for matches.

For a noncorrelated subquery, the inner query is issued once, and the outer query scans the results of the inner query for its matches.

DB2 SQL Performance Analyzer presents information from Explain for both subquery types.

The following example shows a correlated subquery (query 9) and a noncorrelated query (query 10). In the correlated subquery, SQL PA specified that correlation names were found (a correlated subquery block). In the noncorrelated subquery, installation specified limits were exceeded and appropriate messages are displayed.

For each query block, the tables accesses are depicted, along with any chosen indexed paths and other pertinent information, such as index-only processing.

```
EXPLAIN PLAN SET QUERYNO = 100000009 FOR
SELECT * FROM TDT690.S100 X
  WHERE CIKEY BETWEEN 1005 AND 1500
  AND EXISTS (SELECT * FROM TDT690.L1000
               WHERE CIKEY = X.CIKEY)

QUERYNO: 100000009 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 0

CLUSTER MATCH IX SCAN
-----------------------
IX CREATOR: TDT690
INDEX NAME: S100CIN

VERS: 0 KEY LEN: 6 PADDED: - C-ED: N C-ING: N CLUSRATIO: 100.0000
FULLKEY CARD: 839 FIRSTKEY CARD: 839
TYPE: 2 NLEAF PAGES: 4 NLEVELS: 2 UNIQUE: D DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10

KEY COLUMN NAME ORDER TYPE DIST LEN NULL COLCARD DIST#
-----------------------------------------------
1 CIKEY A DECIMAL N 9 Y 839 10

THIS IS THE CLUSTERING ("INSERT & LOAD ORDER") INDEX FOR THIS TABLE
```
INDEXED ACCESS
---------------
CREATOR: TDT690
TABNAME: S100

INDEXED ACCESS IS APPLIED TO A 4K SEGMENTED TSPACE WITH: 3 TABLES
VERSION: 0 TABLE CONTAINS A TOTAL OF: 31680 4K PAGES
TABLE ROWS: 1000 COLUMNS: 10 REC LENGTH: 108 BYTES
DSSIZE: 0 GB NUMBER OF MQTS: 0 LOG: Y AVG ROW LENGTH: 106 BYTES
TYPE: T LOCK SIZE: A TS LOCK MODE: IS LOCK PART: N CLOSE TABLE: Y
PAGES WITH ROWS: 87% PCT COMPRESSED: 0% MAX ROWS: 255 BPOOL: BP11
ENCODE: E CCSIDS ARE SBCS: 833 DBCS: 834 MIXED: 933 VOLATILE: N
THE CORRELATION NAME ASSOCIATED WITH THIS TABLE IS: X

A SUBQUERY HAS BEEN DETECTED. THIS IS QUERY BLOCK NUMBER 2

QUERYNO: 100000009 QBLOCKNO: 2 PLanno: 1 MIXOPSEQ: 0
PROCESS ->

CLUSTER MATCH IX SCAN
----------------------
IX CREATOR: TDT690
INDEX NAME: L1000CIN
VERS: 0 KEY LEN: 6 PADDED: - C-ED: Y C-ING: Y CLUSRATIO: 99.9995
FULLKEY CARD: 99340 FIRSTKEY CARD: 99340
TYPE: 2 NLEAF PAGES: 412 NLEVELS: 3 UNIQUE: D DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10
KEY COLUMN NAME ORDER TYPE DIST LEN NULL COLCARD DIST#
------------------------------------------------------------------------
1 CIKEY A DECIMAL N 9 Y 99340 10

THIS IS THE CLUSTERING ("INSERT & LOAD ORDER") INDEX FOR THIS TABLE

THIS IS AN "INDEX ONLY" ACCESS: NO DATA PAGES ARE READ FROM THE TABLE

THIS IS A CORRELATED SUBQUERY BLOCK.

*---------------------------------------------------------------*
* QUERY 100000009 WILL REQUIRE 125.04680 SECONDS OF ELAPSED TIME *
* DURING WHICH 80.35627 SECONDS OF CPU TIME WILL BE CONSUMED AND *
* A TOTAL OF 10 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK *
* QUNITS 4389 ESTIMATED PROCESSING COST $ 18.3043 DOLLARS *
*---------------------------------------------------------------*

ANL5025W *** WARNING:
ESTIMATE OF 80.356 EXCEEDS "CPU TIME" LIMIT OF 10 CPU SECONDS!

ANL5026W *** WARNING:
ESTIMATE OF 4389 EXCEEDS "SERVICE UNIT" LIMIT OF 200 QUNITS!

ANL5027W *** WARNING:
ESTIMATE OF 18.30 EXCEEDS "MONETARY" LIMIT OF 15.00 DOLLARS!

ANL5029W *** WARNING:
ESTIMATE OF 125.047 EXCEEDS "ELAPSED TIME" LIMIT OF 60 SECONDS!

EXPLAIN PLAN SET QUERYNO = 100000010 FOR
SELECT * FROM TDT690.S100
WHERE CIKEY IN
(SELECT CIKEY FROM TDT690.L1000
WHERE CIKEY BETWEEN 1005 AND 1500)

QUERYNO: 100000010 QBLOCKNO: 1 PLanno: 1 MIXOPSEQ: 0
PROCESS ->
Cluster Match IX Scan

---

IX Creator: TDT690
Index Name: S100CIN

Vers: 0 Key Len: 6 Padded: N C-Ed: N C-Ing: N Clusratio: 100.0000
Fullkey Card: 839 Firstkey Card: 839
Type: 2 NLeaf Pages: 4 NLevels: 2 Unique: D Duplicate OK
1 of 1 Columns Are Matched Close: Y Lock Mode: IS BPool: BP10

Key  Column Name  Order  Type  Dist  Len  Null  ColCard  Dist#
-----------------------------------------------
1  CIKEY  A  DECIMAL  N  9  Y  839  10

This is the clustering ("insert & load order") index for this table

Indexed Access

Creator: TDT690
Tabname: S100

Indexed Access is applied to a 4K segmented tspace with: 3 tables
Version: 0 Table contains a total of: 31680 4K pages
Table Rows: 1000 Columns: 10 Rec Length: 106 Bytes
Dssize: 0 GB Number of MQTs: 0 Log: Y Avg Row Length: 106 Bytes
Type: T Lock Size: A TS Lock Mode: IS Lock Part: N Close Table: Y
Pages with rows: 87% Pct Compressed: 0% Max Rows: 255 BPool: BP11
Encode: E CCSIDs are SBCS: 833 DBCS: 834 Mixed: 933 Volatile: N

A subquery has been detected. This is query block number 1

---

QueryNo: 100000010 QBLOCKNO: 2 PLANNO: 1 MIXOPSEQ: 0
Process ->

Cluster Match IX Scan

---

IX Creator: TDT690
Index Name: L1000CIN

Fullkey Card: 99340 Firstkey Card: 99340
Type: 2 NLeaf Pages: 412 NLevels: 3 Unique: D Duplicate OK
1 of 1 Columns Are Matched Close: Y Lock Mode: IS BPool: BP10

Key  Column Name  Order  Type  Dist  Len  Null  ColCard  Dist#
-----------------------------------------------
1  CIKEY  A  DECIMAL  N  9  Y  99340  10

This is the clustering ("insert & load order") index for this table

This is an "Index Only" access: No data pages are read from the table

---

Anl5025w *** Warning:
Estimate of 557.574 exceeds "CPU time" limit of 10 CPU seconds!

Anl5026w *** Warning:
Estimate of 30448 exceeds "Service Unit" limit of 200 Qunits!
Sorting results

DB2 SQL Performance Analyzer provides information about sorting in its Enhanced Explain Reports, including sorts done on behalf of joins, sorts requested by the user in ORDER BY or GROUP BY clauses, and sorts for uniqueness caused by the DISTINCT or UNION keywords.

The following example shows SQL PA generated information for sorts. Sorts caused by UNION processing or a JOIN of multiple tables might produce several additional messages:

A "UNION" SORT FOR UNIQUENESS HAS BEEN DETECTED
A "JOIN" SORT IS ALSO REQUIRED
A SORT WILL BE PERFORMED ON THE RESULTS

EXPLAIN PLAN SET QUERYNO = 100000011 FOR
SELECT * FROM TDT690.L1000
WHERE CIKEY BETWEEN 501 AND 510
UNION
SELECT * FROM TDT690.L1000
WHERE CIKEY BETWEEN 501 AND 510

QUERYNO: 100000011 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 0
PROCESS ->

CLUSTER MATCH IX SCAN
-----------------------
IX CREATOR: TDT690
INDEX NAME: L1000CIN
VERS: 0 KEY LEN: 6 PADD: - C-ED: Y C-ING: Y CLUSRATIO: 99.9995
FULLKEY CARD: 99340 FIRSTKEY CARD: 99340
TYPE: 2 NLEAF PAGES: 412 NLEVELS: 3 UNIQUE: 0 DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10

++------------------------------------------------------------------++
|ANL6038I *** NOTE: |
|Building an index with the same ordering of sort key columns |
|will negate the sort requested here... consider as warranted. |
ANL7022I *** GUIDELINE:
Avoid referencing Varchar columns in the Sort, as these will be padded to maximum size in the sort work records. Long sort records over 4000 bytes cause a "Tag" sort, which requires even more overhead to park records in a file and read them afterward.

QUERYNO: 100000011 QBLOCKNO: 2 PLANNO: 1 MIXOPSEQ: 0

PROCESS -->
CLUSTER MATCH IX SCAN
-------------
IX CREATOR: TDT690
INDEX NAME: L1000CIN

VERS: 0 KEY LEN: 6 PADDED: - C-ED: Y C-ING: Y CLUSRATIO: 99.9995
FULLKEY CARD: 99340 FIRSTKEY CARD: 99340
TYPE: 2 NLEAF PAGES: 412 MLEVELS: 3 UNIQUE: D DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10

KEY COLUMN NAME ORDER TYPE DIST LEN NULL COLCARD DIST#
------------------------------------------------------------------------
1 CIKEY A DECIMAL N 9 Y 99340 10

THIS IS THE CLUSTERING ("INSERT & LOAD ORDER") INDEX FOR THIS TABLE

INDEXED ACCESS
-------------
CREATOR: TDT690
TABNAME: L1000

INDEXED ACCESS IS APPLIED TO A 4K SEGMENTED TSPACE WITH: 3 TABLES
VERSION: 0 TABLE CONTAINS A TOTAL OF: 31680 4K PAGES
TABLE ROWS: 1000000 COLUMNS: 50 REC LENGTH: 1008 BYTES
DSSIZE: 0 GB NUMBER OF MQTS: 0 LOG: Y AVG ROW LENGTH: 1006 BYTES
TYPE: T LOCK SIZE: A TS LOCK MODE: IS LOCK PART: N CLOSE TABLE: Y
PAGES WITH ROWS: 99% PCT COMPRESSED: 0% MAX ROWS: 255 BPOOL: BP11
ENCODE: E CCSIDS ARE SBCS: 833 DBCS: 834 MIXED: 933 VOLATILE: N

A "UNION" SORT FOR UNIQUENESS HAS BEEN DETECTED.

*------------------------------------------------------------------------*
* QUERY 100000011 WILL REQUIRE 39.55082 SECONDS OF ELAPSED TIME *
* DURING WHICH 34.58685 SECONDS OF CPU TIME WILL BE CONSUMED AND *
* A TOTAL OF 4 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK *
* QUNITS 1889 ESTIMATED PROCESSING COST $ 7.8358 DOLLARS *
*------------------------------------------------------------------------*

ANL5025W *** WARNING:
ESTIMATE OF 34.587 EXCEEDS "CPU TIME" LIMIT OF 10 CPU SECONDS!

ANL5026W *** WARNING:
ESTIMATE OF 1889 EXCEEDS "SERVICE UNIT" LIMIT OF 200 QUNITS!

** Insert, Update, Delete, Select, and Refresh **

DB2 SQL Performance Analyzer reports on INSERT, UPDATE, DELETE, SELECT, and REFRESH statements.
Users might want direct I/U/D processing against the tables, or to use “where current of cursor” techniques. Both forms of requests are handled in the Enhanced Explain reports. Other information, such as the number of columns updated, is also provided.

The DB2 SQL Performance Analyzer Enhanced Explain Report contains descriptions of processing under DB2, with the emphasis on what steps DB2 is taking, and in what order. SQL PA provides the statistics accompanying the tables and indexes used for access. Other notations describe query transformation from subquery to join, subquery to simple predicate, or join to subquery access, by direct rowid.

Referential integrity relationships are also noted in the explain report for INSERT, UPDATE, and DELETE statements, notifying you of the additional processing caused by the relationships.

The following example shows Insert (query 12) processing.

```
EXPLAIN PLAN SET QUERYNO = 100000012 FOR
INSERT INTO TDT690.L1000
VALUES (100100, 'AAAA', 'BBBB', 'CCCC', 'DDDD', 'EEEE', 'FFFF',
'GGGG', 'HHHH', 'IIII', 100100, 'JJJJ', 'KKKK', 'LLLL', 'MMMM',
'NNNN', 'OOOO', 'PPP', 'QQQQ', 'RRRR', 100100, 'SSSS', 'TTTT',
'UUUU', 'VVVV', 'WWWW', 'XXXX', 'YYYY', 'ZZZZ', 'AAAA', 100100,
'BBBB', 'CCCC', 'DDDD', 'EEEE', 'FFFF', 'GGGG', 'HHHH', 'IIII',
'JJJJ', 'KKKK', 'LLLL', 'MMMM', 'NNNN', 'OOOO', 100100, 'PPPP',
'QQQQ', 'RRRR', 'SSSS')
```

**INSERT VIA CLUSTERING INDEX**

```
CREATOR: TDT690
TABNAME: L1000
VERS: 0 KEY LEN: 6 PADDED: - C-ED: Y C-ING: Y CLUSRATIO: 99.9995
FULLKEY CARD: 99340 FIRSTKEY CARD: 99340
TYPE: 2 NLEAF PAGES: 412 NLEVELS: 3 UNIQUE: 0 DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IS BPOOL: BP10
```

|ANL7012I***GUIDELINE: Make sure that you have defined adequate Free Space parms in this index structure, and in the data table, to keep rows in the clustered order and avoid fragmentation. Both Freepage and Pctfree should be deployed, and Runstats periodically run to update stats.|

**INSERT VIA IDX**

```
CREATOR: TDT690
TABNAME: L1000
INSERT PROCESSING APPLIED TO A 4K SEGMENTED TSPACE WITH 3 TABLES
VERSION: 0 TABLE CONTAINS A TOTAL OF: 31680 4K PAGES
TABLE ROWS: 100000 COLUMNS: 50 REC LENGTH: 1008 BYTES
DSSIZE: 0 GB NUMBER OF MQTS: 0 LOG: Y AVG ROW LENGTH: 1006 BYTES
TYPE: T LOCK SIZE: A TS LOCK MODE: IX LOCK PART: N CLOSE TABLE: Y
PAGES WITH ROWS: 99% PCT COMPRESSED: 0% MAX ROWS: 255 BPOOL: BP11
ENCODE: E CCSIDS ARE SBCS: 833 DBCS: 834 MIXED: 933 VOLATILE: N
```

* QUERY 100000012 WILL REQUIRE 28.69939 SECONDS OF ELAPSED TIME *
* DURING WHICH 25.63702 SECONDS OF CPU TIME WILL BE CONSUMED AND *
The following example shows Update (query 13) processing.

```
EXPLAIN PLAN SET QUERYNO = 100000013 FOR
UPDATE TDT690.L1000
  SET C2 = 'NEW11', C7 = 7654321
  WHERE CIKEY = 100100

QUERYNO: 100000013  QBLOCKNO: 1  PLANNO: 1  MIXOPSEQ: 0
PROCESS ->
```

**UPDATE OPERATION HAS BEEN REQUESTED TO CHANGE THE VALUE OF 2 COLUMNS**

```
+------------------------------------------------------------------+
|ANL7001I ***GUIDELINE:|
|Make sure that your frequently updated columns are being kept together in the table row: DB2 logs from the first to last changed byte. Also, put Varchar at the end of the row: DB2 logs from the first changed byte to end of row. Length changes cause the entire row to log, as length field is kept with the row header. Rows that are compressed are variable length & logged in compressed format.
+------------------------------------------------------------------+
```

**CLUSTER MATCH IX SCAN**

```
-------------
IX CREATOR: TDT690
INDEX NAME: L1000CIN
```

```
VERS: 0  KEY LEN: 6  PADDED: -  C-ED: Y  C-ING: Y  CLUSRATIO: 99.9995
FULLKEY CARD: 99340  FIRSTKEY CARD: 99340
TYPE: 2  NLEAF PAGES: 412  NLEVELS: 3  UNIQUE: D  DUPLICATE OK
1 OF 1 COLUMNS ARE MATCHED  CLOSE: Y  LOCK MODE: IX  BPOOL: BP10

KEY COLUMN NAME  ORDER  TYPE  DIST  LEN  NULL  COLCARD  DIST#
-------------  --------  ----  ----  ----  ----  ------  ----#
1  CIKEY  A  DECIMAL  N  9  Y  99340  10
```

**THIS IS THE CLUSTERING ("INSERT & LOAD ORDER") INDEX FOR THIS TABLE**

```
+------------------------------------------------------------------+
|ANL6020I ***NOTE:|
|This index is presently designated as allowing "duplicate values". Make sure this is a nonunique index that does not fall into one of the following "unique" categories: explicitly unique, primary key, non-primary RI parent key, unique where not null, unique column constraint. Unique indexes have definite advantages whenever they can be declared by the user, so be sure it contains duplicates.
+------------------------------------------------------------------+
```

**THIS IS AN "INDEX ONLY" ACCESS: NO DATA PAGES ARE READ FROM THE TABLE**

**UPDATE VIA IDX**

```
-------------
CREATOR: TDT690
TABNAME: L1000
```

**UPDATE PROCESSING APPLIED TO A 4K SEGMENTED TSPACE WITH 3 TABLES**

```
VERSION: 0  TABLE CONTAINS A TOTAL OF: 31680 4K PAGES
TABLE ROWS: 100000  COLUMNS: 50  REC LENGTH: 1008 BYTES
```
The following example shows Delete (query 14) processing.

```
EXPLAIN PLAN SET QUERYNO = 100000014 FOR
DELETE FROM TDT690.L1000
WHERE CIKEY = 100100 OR RIKEY2 = 100300

QUERYNO: 100000014 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 0
PROCESS ->

A MULTIPLE INDEX OPERATION HAS BEEN REQUESTED TO ACCESS THIS TABLE:

QUERYNO: 100000014 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 1
PROCESS ->

RANDOM MATCH IX SCAN
----------------------
IX CREATOR: TDT690
INDEX NAME: L1000P2N
VERS: 0 KEY LEN: 6 PADDED: N C-ED: N C-ING: N CLUSRATIO: 70.0010
FULLKEY CARD: 100000 FIRSTKEY CARD: 100000
TYPE: 2 NLEAF PAGES: 359 NLEVELS: 3 UNIQUE: U DECLARE UNIQ
1 OF 1 COLUMNS ARE MATCHED CLOSE: Y LOCK MODE: IX BPOOL: BP10

+------------------------------------------------------------------+
|ANL6052I *** NOTE: |
|This index is specified as "Not Padded", allowing storage of a |
|varying length index key. Padded indexes use blanks to fill out |
|their fixed length keys and are not eligible for Index Only scan. |
|"Not Padded" indexes do not blank fill CHAR and VARCHAR columns, |
|allowing greater flexibility and better use of storage, packing |
|more entries into a single Leaf page. Index Only access allowed. |
+------------------------------------------------------------------+

KEY COLUMN NAME ORDER TYPE DIST LEN NULL COLCARD DIST#
------------------------------------------------------------------------
1 RIKEY2 A DECIMAL N 9 Y 100000 0

THIS IS AN "INDEX ONLY" ACCESS: NO DATA PAGES ARE READ FROM THE TABLE

QUERYNO: 100000014 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 2
PROCESS ->

CLUSTER MATCH IX SCAN
---------------------
IX CREATOR: TDT690
```
**DB2 SQL Performance Analyzer detail trace report**

The DB2 SQL Performance Analyzer detail trace report, which is the most detailed, contains hypothetical trace information covering virtually every facet of processing.

The information describes each query step as though it were issued, and shows the expected usage of all system resources. DB2 logical entities are isolated, with the processing broken down into small, finite pieces. The individual components contributing to overall response time is shown, allowing you to fine-tune queries at the most granular level possible.

For example, DB2 SQL Performance Analyzer shows the predicted path lengths in instructions for fetching rows, or performing sequential prefetch I/O calls, or logging records updated by the query. SQL PA also predicts how long DB2 spends scanning the rows. The elapsed (response) time is broken down into its components, from waiting for data sets to open and close, to waiting for synchronous (or prefetch) I/O to complete.

The report is written to the file allocated by the QTRACE DD statement in batch and to the file designated as the trace report in the SQL PA Define Report Data Sets panel.

DB2 SQL Performance Analyzer shows its predictions in typical DB2 terms, such as forecast numbers for getpages, synchronous reads, sequential and list prefetches, and the CPU time users might anticipate in the DB2 SMF records.

**Topics:**

- “Forecasting query processing” on page 207
- “List of the DB2 SQL Performance Analyzer install parameters” on page 207
Forecasting query processing

The entire SQL PA cost summary estimate, along with the forecasted elapsed time, CPU time, and I/O are determined by mapping the access path chosen by DB2 against the current picture of the tables and indexes, as represented by the catalog.

Using known physical attributes for CPU processing speeds, disk device service times, and thousands of individual path lengths observed using private benchmarks of DB2, the query execution is analyzed.

DB2 SQL Performance Analyzer parses the SQL, and detects the type of query. SQL PA analyzes as much as possible for the cost estimation. The cost estimation is based on all the small and large details that influence performance, including the calculations on a WHERE clause, built-in functions, the use of certain predicates, join table orders, and RID list counts.

**DB2 SQL Performance Analyzer standard Cost Summary format:**

```
* QUERY 100000007 WILL REQUIRE 1.79174 SECONDS OF ELAPSED TIME *
* DURING WHICH 0.13270 SECONDS OF CPU TIME WILL BE CONSUMED AND *
* A TOTAL OF 5 PHYSICAL I/O REQUESTS WILL BE ISSUED TO DISK *
* QUNITS 2  ESTIMATED PROCESSING COST $ 0.2755 DOLLARS *
```

**DB2 optimizer’s Cost Summary format (PRECISE YES):**

```
* QUERY: 100000003 HAS BEEN ESTIMATED BY THE DB2 OPTIMIZER *
* TO CONSUME 9.24109 SECONDS OF CENTRAL PROCESSING TIME *
* WHICH IS EQUIVALENT TO 471 RESOURCE SERVICE UNITS. *
* ESTIMATED COST OF CPU PROCESSING: $ 1.2833 DOLLARS *
```

The key statistical variables presented in the SQL PA Detail Trace Report are mapped to their DB2 counterparts in the sample Detail Trace Report.

**List of the DB2 SQL Performance Analyzer install parameters**

The Detail Trace report lists all of the DB2 SQL Performance Analyzer installation parameters at the top of the report.

Unless DSPPARM is set to NO, the beginning of a Detail Trace report lists the key parameter values used by DB2 SQL Performance Analyzer for a run, as shown in the following example:

```
15:23:50.338 SQL Performance Analyzer Version 4.2.0
06-25-2013 Execution Forecast: Detail Trace Level 42-4200 APAR PK73949
```
SQL PA Parameters

(ANLPARM)
COMMENT **********************************************************************
COMMENT >>>>>> PARAMETERS DISCOVERED DURING INSTALL <<<<<<<<<<<<<<<<<
COMMENT **********************************************************************
BUFFHIT 99.59 0% OF PAGES ARE FOUND IN BUFFER POOL
COMMENT **********************************************************************
COMMENT >>>>>> SET THE RUN TIME ENVIRONMENT <<<<<<<<<<<<<<<<<<<
COMMENT **********************************************************************
COMMENT >>>>>> OPERATIONAL PARAMETERS <<<<<<<<<<<<<<<<<<<<<<<<<<<
COMMENT **********************************************************************
CONNECT NONE ATTACH OVERHEAD (CAF|DRDA|CICS|RRSAF|NONE, ETC.)
REFRESH ANY CONSIDER V8 MQTS FOR ACCESS PATHS (YES | ANY | NO)
VIADRDA +OFF+ ROUTE THIS SQL TO A REMOTE DB2 LOCATION
RETCODE NO LET RETCODE REFLECT MESSAGING LEVELS (YES|NO)
DELIMIT +OFF+ ONLY COBOL QUOTESQL OPTION REQUIRES (QUOTE | +OFF+)
DSNBEXP NO EXPLAIN USING DB2 SAMPLE STORED PROCEDURE (YES|NO)
NLSCODE ON MIXED/DBCS SUPPORT VALUES (KOR|JPN|CHT|CHS|ON|+OFF+)
KEEPLAN NO RETAIN THE CONTENTS OF PERSONAL PLAN TABLE (YES|NO)
PROCESS +OFF+ NOSEQ IS FOR DBRMS WHICH HAVE STMT#S OUT OF SEQUENCE
NUMBERS YES YES | NO IGNORE LAST 8 CHARS FROM ANLIN SEQ FILE
COMMENT **********************************************************************
COMMENT >>>>>> REPORTING PREFERENCES <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
COMMENT **********************************************************************
REPORTS DET CHOOSE LEVEL: LOG, REP, DET
SHOWALT YES SHOW ALTERNATE ACCESS PATHS NOT CHOSEN (YES|NO)
PRECISE ALL EXTRA COST OPTIMIZATION FOR PRECISION (YES|NO|ALL)
ADVISOR ALL WARNINGS/ALERTS + RECS/NOTES=YES, GUIDES/GOOD=ALL
OBJECTS YES PTF3104 SHOW ALL TABLES AND INDEXES ACCESSED BY SQL
QLIMSQL YES SHOW SQL THAT FAILED THE EXPLAIN PREPARE (QLIMITS)
MESSAGE YES IF NO, THEN NO ADVISOR MESSAGES PRINT (JUST MSGIDS)
NOTSCAN NO IF YES, EVERY TABLESPACE SCAN WILL GET MESSAGE 5042W
NOSTATS YES ISSUE WARNING IF NO STATS FOR OBJECTS, MESSAGE 3026W
ALLPART YES ISSUE WARNING ANL5044W IF ALL PARTITIONS ARE TSCANED

208 User's Guide
NONONIX NO IF YES, EVERY NONMATCH INDEX WILL GET MESSAGE 5043W
TURNOFF 9999 USED TO 'TURNOFF' MESSAGE IDS FROM THE ADVISOR LIST
DSPVARS NO YES DISPLAY HOST VARS, NO DISPLAY PARAMETER MarkERS
DSPPARM YES YES | NO DISPLAY PARAMETERS USED ON TOP OF REPORTS
COMMENT ***************************************************************
COMMENT >>>>>> SET LIMITS FOR REPORTING MESSAGES WHEN LIMIT IS EXCEEDED
COMMENT ***************************************************************
JOIN TAB 10 LIMIT TO NUMBER OF JOIN TABLES BEFORE MESSAGE 7017I
INLISTS 10 LIMIT NUMBER ELEMENTS IN(LIST) BEFORE MESSAGE 5007W
IXUPDAT 5 LIMIT NUMBER INDEXES ON UPDATE BEFORE MESSAGE 6001I
TSCANPG 50000 NUMBER OF PAGES READ BY TSCAN BEFORE MESSAGE 6066I
ISCANPG 10000 NUM LEAF PAGES IN MATCH IX SCAN BEFORE MESSAGE 6067I
NONIXPG 50000 NUM LEAF PAGES IN NONMATCH SCAN BEFORE MESSAGE 6068I
MATCOLS 0.50 LIMIT PERCENT COLUMNS MATCHED IN SCAN, MESSAGE 606II
PREDPCT 0.15 LIMIT PCT ROWS FILTERED BY PREDICATES, MESSAGE 6025I
COMMENT ***************************************************************
COMMENT >>>>>> PERCENTAGE OF PAGES <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
COMMENT ***************************************************************
COMMENT >>>>>> DEVICE TYPE FOUND IN THE USERS GUIDE APPENDIX C <<<<<<<<
COMMENT ***************************************************************
STORAGE ESS800 THE DASD FARM IS IBM 'SHARK' ESS SOLID STATE DRIVES
(ANLPARM)
DEGREES 1
DBRMKEY DBRNAME
EXPLAIN NEW
EEECALL SPA
QUALIFY PROD
USEPLAN PROD
QUERY NO 1
STOPQNO 999999999
VERSION N/A will be discovered by code
(ANLCNTL)
COMMENT ***************************************************************
COMMENT >>>>>> PARAMETERS DISCOVERED DURING INSTALL <<<<<<<<<<<<<<<<<<<<<

Chapter 6. DB2 SQL Performance Analyzer reporting capabilities 209
COMMENT ***************************************************************

ENGINES 10 ENGINES
BUFFERS 00020000 SIZE OF 4K APPLICATION BUFFER POOL
BUFF08K 00000500 8K BUFFER POOLS FOR APPL
BUFF16K 00000250 16K BUFFER POOLS FOR APPL
BUFF32K 00000100 SIZE OF 32K APPLICATION BUFFER POOL
DYNAMIC YES CACHING OF DYNAMIC SQL STATEMENTS (YES\|NO)
COMMENT ***************************************************************

COMMENT >>>>>> SET THE RUN TIME ENVIRONMENT <<<<<<<<<<<<<<<<<<<<<<<<<<<
COMMENT ***************************************************************

DBTRACE OFF DIAGNOSTICS TRACE PARM: OFF | ON | ALL | DMP
SETPLAN YES AUTHORIZE USER TO SPECIFY HLQ FOR PLAN_TABLE
ANLKEYS 00000 PARSER=01000 CLASS1=10000 NOVER=02000 NOPT=20000
COMMENT ***************************************************************

COMMENT >>>>>> HARDWARE/SOFTWARE ENVIRONMENT <<<<<<<<<<<<<<<<<<<<<<<<<<<
COMMENT ***************************************************************

LPARENG 01.000 PORTION OF THE BOX DEDICATED TO DB2
ESASORT YES HARDWARE ASSISTED SORT IS ACTIVE
ESACOMP YES HARDWARE ASSISTED DATA COMPRESSION IS ACTIVE
COMMENT ***************************************************************

COMMENT >>>>>> ADJUST THE CPU OVERHEAD IN A DATA SHARING ENV <<<<<<<<
COMMENT ***************************************************************

DSGROUP 0 # OF MEMBERS IN THE DATA SHARING GROUP, 0-32
DATASHR 0 AVG % WORKLOAD PARTICIPATES IN DATA SHARING,0-100
COMMENT ***************************************************************

COMMENT >>>>>> SET COST LIMITS USED FOR WARNING MESSAGES <<<<<<<<<<<
COMMENT ***************************************************************

CPUTIME 00010 10 SECS OF CPU TIME
COSTING 00000015 $15.00 QUERY COST (MONETARY TERMS)
ELAPSED 00060 1 MINUTE ELAPSED TIME
COSTQUN 00000200 200 QUERY SERVICE QUNITS
IOCALLS 00001000 1000 I/O CALLS DISK I\O (PHYSICAL)

COMMENT ***************************************************************
**Processing steps**

Each major step in the DB2 process noted in the Enhanced Explain Report is expanded upon in the Detail Trace Report. The SQL PA Detail Trace Report presents an estimate of how many rows and columns are processed by the statement, along with the percent of the table processed.

These process estimate numbers are derived by SQL PA as the DB2 optimizer would derive them, and are based on WHERE clause predicates and their filter factors. At times, they are different from what the user experiences because the optimizer uses predefined filters for many of the predicates, and defaults for others.

The OPTIMIZE FOR \( n \) ROWS clause and FETCH FIRST \( n \) ROWS ONLY are recognized by DB2 SQL Performance Analyzer as an attempt to establish a more precise estimate based on rows returned. When DB2 SQL Performance Analyzer detects this clause, it does not use the optimizer emulation calculations, but starts with your actual row estimate for greater accuracy. Give SQL PA a realistic estimate of the actual number of rows returned by a query for the best results. From this estimate, SQL PA calculates many things, including the number of pages needed to hold those rows, the I/O required for those pages, the processing impacts.

You can use the internal estimates of the optimizer for service units and CPU time using the PRECISE parameter (see "ANLPARM user parameters" on page 267). However, the estimates of the optimizer might not be as accurate as a row estimate that you provide.
The banner for each SQL statement contains QUER YNO (query number), QBOCKNO (query block number), PLANNO (plan number) and MIXSEQ (multiple index operational sequence) from the DB2 Explain facility which identify each step in the process.

The block is marked PROCESS -> and contains the following estimates:

**ROWS PROCESSED**
DB2 SQL Performance Analyzer estimate of rows returned, based on predicate filters or OPTIMIZE FOR n ROWS clause or FETCH FIRST n ROWS clause.

**PERCENT TABLE PROCESSED**
DB2 SQL Performance Analyzer calculated percentage of the table accessed by query (always computed).

**COLS PROCESSED**
A count of columns from SELECT list, or from catalog if all rows selected.

**BOOLEAN FILTER FACTORED** or **BOOLEAN FILTER FROM DB2|OPT**
DB2 SQL Performance Analyzer final filter factor from predicate analysis (or from CPU time as supplied by optimizer).

**DATA PAGES READ**
Estimate of total table page reads required by this PLAN/MIXSEQ step.

**INDEX LEAF PAGES READ**
DB2 SQL Performance Analyzer estimate of index leaf page reads; it does not include root or nonleaf pages.

Each major step in the DB2 process noted in the Enhanced Explain Report is expanded upon in the Detail Trace Report.

For each step, there is a summary which includes the enhanced explain information, and estimates for many key variables and path lengths in the DB2 SQL Performance Analyzer evaluation process. These estimates are computed by SQL PA. The estimates are based on the DB2-explained plan, the catalog statistics for the indexes and tables used in this step of the plan, optimizer algorithm knowledge, and SQL PA parser intelligence data. The step information in the Detail Trace Report contains information like the following example:

```sql
SELECT * FROM TDT690.L1000
   WHERE CIKEY = 50086
```

<table>
<thead>
<tr>
<th>QUERYNO: 100000001</th>
<th>QBOCKNO: 1</th>
<th>PLANNO: 1</th>
<th>MIXSEQ: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS -&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
+--------------------------------------------------------+
| ANL7902I *** GUIDELINE: |
| This plan step has not selected any Sequential|List Prefetch I/O. |
| If the SQL processes just a few rows that is OK, but if many rows |
| are involved, you can help promote Sequential Detection by both |
| accessing the data in sequential order (presort?) and by binding |
| with Release (Deallocate) to avoid resetting counters at Commit. |
+--------------------------------------------------------+
```

```
+--------------------------------------------------------+
| ANL7903I *** GUIDELINE: |
| Close Yes was specified for the Tablespace and/or the index... |
| if these are little used this is OK. If high volume access then |
| consider Close No. Extremely relevant pages can be "page fixed" |
| in memory by highly referencing, using Hiperpools (Castout Y/N), |
| putting into a dedicated buffer pool large enough to hold all |
+--------------------------------------------------------+
```
pages, deploying data sharing with Group Buffer Pool Cache All option, etc. each with associated costs. Close No also increases chances that DBD will remain in EDM Pool for next execution.

ANL7006I ***
This statement contains a select of all columns in the table. Typically a select of all columns results in significantly increased processing time to fetch and process each column of each row. Use select all only when you want to select all columns in a view definition. Improve performance by specifying the name of each column for the SQL SELECT statement.

ANL6020I *** NOTE:
This index is presently designated as allowing "duplicate values". Make sure this is a nonunique index that does not fall into one of the following "unique" categories: explicitly unique, primary key, non-primary RI parent key, unique where not null, unique column constraint. Unique indexes have definite advantages whenever they can be declared by the user, so be sure it contains duplicates.

ANL7015I *** GUIDELINE:
Collecting Non-uniform column statistics greatly enhance the DB2 Optimizer accuracy when processing Equals and In (List) preds.

RowS Processed = 10 Percent Table Processed = .000100000
Cols Processed = 50 Boolean Filter From Opt = .000100000
Data Pages Read = 3 Indexed Leaf Pages Read = 1
Sync Read I/OS = 6 Table = 3 Index = 3
Prefetch I/OS = 0 Table = 0 Index = 0
Async Wrt I/OS = 0 Table = 0 Index = 0
Get Page Calls = 38 Sysio = 32 Logio = 0
SYNC READ = 0.044100  PREFETCH = 0.000000  ASYNC WRT = 0.000000
GET PAGES = 0.093100  SYSTEMS = 0.258450  LOG WRITE = 0.000000
DECOMPRESS= 0.000000  COMPRESS = 0.000000  HIPERPOOL = 0.000000
FETCH ROW = 0.044500  LOCK/ETC = 0.042000  PROCESSES = 0.066795

PREDICATE = 1  CLASS 1 = 0.548945  OTHER O/H = 0.846400
QUERYNO: 100000001  QBLOCKNO: 1  PLANNO: 1  MIXOPSEQ: 0

SUMMARY ->

THE TOTAL CUMULATIVE PATH LENGTH FOR THIS QUERY IS 1.395345M INS.
RESULTING IN A TOTAL CPU TIME OF 0.01311 SECONDS CONSUMED OVERALL.
DB2 WILL PUT CLASS 1 CPU TIME OF 0.00516 SECONDS IN SMF 101 RECORD.
DB2 SHOWS ADDITIONAL CPU TIME OF 0.00795 SECONDS IN SMF 100 RECORD.
ESTIMATED TOTAL LOGICAL I/O CALLS = 6 (EXCLUDING SYSTEM) AND
ESTIMATED TOTAL PHYSICAL I/O CALLS = 6 WITH HIT RATIO = 1.000.
WAIT TIME FOR SYNC READ I/O = 0.13831  PREFETCH I/O = 0.000000
WAIT TIME ON ASYNC WRITE I/O = 0.000000  TOTAL IWAIT = 0.13831
WAIT TIME FOR VSAM OPEN/CLOSE MACROS, BINDING AND LOCKING = 1.79730

******************************************************************************

I/O estimates

The DB2 SQL Performance Analyzer detail trace report includes I/O estimate information. The total number of data pages and index leaf pages read is a function of the predicate filters and the SQL PA estimate of the percent of the table and index processed to retrieve the necessary number of rows.

Some of these pages are read through synchronous reads (one page at a time) or by sequential prefetch (both table and index) or list prefetch (table only). Thus, the number of pages does not necessarily equal the number of I/Os. An asynchronous sequential or list prefetch can read in 8, 16, 32, or more pages with a single I/O request.

DB2 SQL Performance Analyzer separates the individual I/O estimates in the SQL PA Detail Trace Report as follows:

SYNC READ I/Os
DB2 SQL Performance Analyzer estimate of the total number of single, synchronous read I/Os issued; separate counts are provided for TABLE and INDEX.

PREFETCH I/Os
DB2 SQL Performance Analyzer estimate of the total number of multiple page, asynchronous read I/Os issued, including both sequential and list prefetches; TABLE and INDEX counts are again provided.

ASYNC WRT I/Os
DB2 SQL Performance Analyzer estimate of the asynchronous write I/Os caused by INSERT, UPDATE, or DELETE statements; the number of pages written back to disk with updated values for TABLE and INDEX pages.

GET PAGE CALLS
Fully loaded estimate of getpage calls, which includes all the logical reads for tables and indexes, and the system I/O (see SYSIO).
SYSIO

Estimate of internal getpages caused by actions such as dynamic binds, as appropriate for query.

LOGIO

DB2 SQL Performance Analyzer estimate of DB2 log writes in support of insert, update, and delete transactions, if any.

The logical I/O numbers do not necessarily equal the total number of physical I/O calls caused by this step in the query. Consider that the logical I/O requests are uniform for each query. If a query needs to read 25 data pages and 6 index pages, then it requires the same page counts (getpages) every time, unless the predicates or data composition change. Whether those pages are already found in the buffer pool, or need to brought in from disk, determines the buffer hit ratio. Physical I/O consists of those pages which must be read from disk, and not the total getpages issued. The parameter BUFFHIT, reflects this physical to logical I/O ratio, and all three values are shown later in the report.

Path lengths and CPU time

The Detail Trace Report includes the path lengths, or instruction counts, for various processes during the processing of each step.

DB2 SQL Performance Analyzer obtains the information through a series of benchmarks that capture the individual costs of each DB2 process, and are like most analytical queuing and discrete simulation models used for DB2-related capacity planning. The path lengths refer to the number of instructions needed to accomplish a specific task within DB2, and are the results of thousands of benchmarked measurements on all possible DB2 access paths and scenarios.

The path lengths are presented at a granular level, to provide you with an incremental-cost view of each query processed. In the report, you are able to see where SQL PA estimates the bulk of processing time is to be spent. All path lengths are presented in millions of instructions, so a value of 0.950750 represents 950,750 instructions, and a value of 23.500250 indicates that 23,500,250 instructions are issued.

DB2 SQL Performance Analyzer is not a modeling tool, but it can provide input about individual transaction resource consumption to a modeling tool. SQL PA provides a complete, end-to-end picture of individual query processing. These are the building blocks of the capacity model, which then summarize the individual queries, combining them into workloads at various arrival rates. These workloads are then used to project the overall application performance and capacity of a DB2 system.

DB2 SQL Performance Analyzer does not predict performance of specific application volumes or within specific workload mixes. Application performance is a job reserved for the queue and simulation model tools. The role of the model is to evaluate the performance of the entire application, given a specific transaction mix, while considering the rest of the workloads processing on the complex.

The role of DB2 SQL Performance Analyzer is to evaluate the performance of each individual query. SQL PA does predict individual transaction performance, and evaluates the relative cost of using one method to access data compared with another. With the performance information provided by SQL PA, you can determine individual query performance assessment and perform fine-tuning of your queries.
Path length-related values shown in the report include the following categories:

**PROCESSES**
Estimate of the primary processing path length, including Stage 1 and Stage 2 predicate scans, index and data page scans, built-in functions, calculations, and similar processes for the access path chosen.

**CLASS 1**
DB2 SQL Performance Analyzer estimate of those processes charged to class 1 CPU time, including PROCESSES, GET PAGE, FETCH ROW, SYNC READ, SYSTEMS, DECOMPR, LOCK/ETC, and the Read part of HIPERPOOL.

**OTHER O/H**
DB2 SQL Performance Analyzer estimate of those processes charged to DB2 processor usage, including the Attach facility, PREFETCH, LOG WRITE, COMPRESS, and ASYNCWRT.

**SYNC READ**
Estimate for the cost of issuing all single page, synchronous read I/O requests.

**PREFETCH**
DB2 SQL Performance Analyzer estimate for the cost of issuing all multi-page sequential and list prefetch I/O requests.

**ASYNC WRT**
DB2 SQL Performance Analyzer estimate for the cost of issuing all data and index page writes to disk, after inserts, updates, and deletes.

**DECOMPRESS**
The path length to decode rows for processing.

**COMPRESS**
The path length to encode rows before write.

**HIPERPOOL**
The total path length for hiperpool activity, including both Reads and Writes of updated pages.

**GET PAGES**
DB2 SQL Performance Analyzer estimate for the cost of all getpage logical I/O requests issued by this step.

**SYSTEMS**
General processor usage initiated by the system for the SQL statement, including open and close table, index and cursor, and commits.

**LOG WRITE**
The cost estimate for any log records that are written because of inserts, updates, and deletes by this step of the query.

**FETCH ROW**
Estimate for the path length caused by fetching the rows and columns for this step.

**LOCK/ETC**
DB2 SQL Performance Analyzer category for the odd path lengths, in which locking, HAVING clauses, descending sort and other such path length costs are attributed.

**PREDICATE**
The number of predicates detected on the WHERE clause by the parser.
The two summary fields, **CLASS 1** and **OTHER O/H**, are totals of other fields which appear in this section.

```plaintext
CLASS 1 = PROCESSES + SYNC READ + GET PAGES +
          SYSTEMS + FETCH ROWS + LOCK/ETC +
          HIPERPOOL (Reads) + DECOMPRESS

OTHER O/H = PREFETCH + ASYNC WRT + LOG WRITE +
            COMPRESS + HIPERPOOL (Writes)
```

All numbers are given in millions of instructions.

**TOTAL CUMULATIVE PATH LENGTH**

A running total of the entire path length for all plan or multiple index operational sequence (MIXSEQ) steps in the process.

To create a model for estimating a transaction that will be run repeatedly in a bundle, do not multiply the result of one statement by the number of times that the statement will be run. Since the connection overhead is only needed once, multiplying the running total for one will overstate the results. Instead, remove the processor usage before you multiply, and add the processor usage to the result after you multiply, if the transactions can be bundled.

The next section of the report, following the cumulative path length, converts these path lengths into CPU time, based on the description of the CPU processor specified in the installation parameters.

The following example shows the cumulative path length and CPU time estimates, along with I/O and wait time values.

```
QUERYNO: 100000001 QBLOCKNO: 1 PLANNO: 1 MIXOPSEQ: 0
SUMMARY ->

THE TOTAL CUMULATIVE PATH LENGTH FOR THIS QUERY IS 1.395345M INS.
RESULTING IN A TOTAL CPU TIME OF 0.01311 SECONDS CONSUMED OVERALL.
DB2 WILL PUT CLASS 1 CPU TIME OF 0.00516 SECONDS IN SMF 101 RECORD.
DB2 SHOWS ADDITIONAL CPU TIME OF 0.00795 SECONDS IN SMF 100 RECORD.
ESTIMATED TOTAL LOGICAL I/O CALLS = 6 (EXCLUDING SYSTEM) AND
ESTIMATED TOTAL PHYSICAL I/O CALLS = 6 WITH HIT RATIO = 1.000.
WAIT TIME FOR SYNC READ I/O = 0.13831 PREFETCH I/O = 0.00000
WAIT TIME ON ASYNC WRITE I/O = 0.00000 TOTAL IWAIT = 0.13831
WAIT TIME FOR VSAM OPEN/CLOSE MACROS, BINDING AND LOCKING = 1.79730
```

Figure 54. Process totals

The total cumulative path length for this query contains the rolling cumulative total of all instructions issued, including the **CLASS 1**, and **OTHER O/H** path lengths assessed for each step. The value in this field is typically a cumulative total, except in the following two cases—multiple index operations and subqueries. In these two situations, the value shows the individual path length for the multiple index operation step or the subquery block, providing you with a greater insight into the incremental costs of the query.

Resulting in a total CPU time of xx seconds consumed overall converts those instruction counts into CPU time, using the processor speed determined by the
configuration parameters. Regardless of whether the path length is cumulative or individual (MIX, SBQRY), the CPU times calculated in this total are always cumulative.

DB2 puts class 1 CPU time of xx seconds in SMF 101 record shows the part of the CPU time that SQL PA estimates shows up on a class 1 accounting report; that is, charged to the user. This class 1 estimate is part of the total CPU time just above it on the trace report, and corresponds to the total CLASS 1 path length.

DB2 shows additional CPU time of xx seconds in SMF 100 record shows the CPU time that SQL PA estimates are collected system wide and put in statistics record; that is, absorbed by the DB2 address spaces. Additional CPU time corresponds to CPU time consumed by the OTHER O/H path length, and is the other part of the total CPU time two lines above it.

**Physical and logical I/O**

SQL PA reports estimates of the logical I/O requirements of a query and the physical I/O consequences of those requests.

There is a substantial difference between the logical I/O requirements of a query and the physical I/O consequences of those requests. SQL PA reports both estimates, governed by the BUFFHIT parameter, which specifies the percentage of time that a getpage is satisfied by a page found in the buffer pool. This is important for determining overall elapsed time and query response because the wait time for various physical I/O is a significant contributor to the elapsed time estimate. Similarly, HPOOLRD influences the Physical I/O elapsed time—the more pages read from hiperpool, the less time to read.

**Estimated total logical I/O calls**

The anticipated I/O requirements of the query. The total is based on the filter factors derived from the predicates, and the number of index and data table pages SQL PA has forecasted to be read to resolve the query. This number should not change between processing of identical queries. Logical I/O includes all synchronous reads, sequential and list prefetch requests, asynchronous writes, logging, sort work reads and writes, and those special scans caused by subqueries and joins.

**Estimated total physical I/O calls** are what is left of the logical I/O calls after applying the BUFFHIT parameter to them. SQL PA expects these I/O to require disk access, and these pages must be brought into the buffer pool before scanning. This number can vary between processing of identical queries, as the number of pages found in the buffer pool could vary with usage by many users. Some of these pages can also be retrieved from hiperpool (HPOOLRD) rather than disk.

**With hit ratio**

Describes the physical I/O factor for this run or Logical I/O * (1 - BUFFHIT) = Physical I/O. The elapsed time estimate includes physical I/O wait time. SQL PA computes elapsed time with the knowledge of which pages are read from disk as opposed to those pages found in the buffer pool. Thus, elapsed times predicted by SQL PA vary with the BUFFHIT parameter value.
Elapsed wait time

The wait time for physical I/O is one of the components of elapsed time. The Detail Trace Report tracks separate wait time estimates for the following major I/O categories.

The BUFFHIT, DEGREES, STORAGE, VERSION, and CONNECT user parameters, with the system configuration description parameters in ANLCNTL influence the outcome of DB2 SQL Performance Analyzer elapsed time estimation.

Similarly, defining your own DASD devices with the NEWSTOR, NEWSEEK, NEWROTA, and NEWXFER parameters can dramatically change the elapsed time estimates.

Wait time for sync read I/O

Shows the calculated amount of time to process the estimated number of synchronous reads, using the disk characteristics discussed in The primary DB2 SQL Performance Analyzer user parameters. Specifically, the time to read a 4 KB, 8 KB, 16 KB, or 32 KB page is calculated and shown. The wait time varies with different disk devices specified as STORAGE, because they operate at varying speeds.

Wait time for prefetch I/O

Shows the calculated amount of time to process the estimated number of I/Os for sequential and list prefetch activities. The physical disk characteristics are applied to the number of I/O estimated by DB2 SQL Performance Analyzer. Hiperpool reads are considered here, and any parallel I/O opportunities, and their effect on elapsed time.

Wait time for async write I/O

Shows wait time for actually updating the data and index pages on disk, typically a process performed after commit (and often after user termination). The values are normally not seen on a class 1 report. However, they are an important part of the workload for inserts, updates, and deletes. The values are ascertained by DB2 SQL Performance Analyzer. Hiperpool writes are also considered here.

Total IWAIT time

Shows an aggregate summation of the Wait times for Sync Read, Prefetch and Async Write, after applying the BUFFHIT hit ratio. It determines the anticipated wait time only for I/O that is physically taking place, and where it is happening (hiperpool or disk).

Wait time for VSAM open/close, binding, locking

Collects the remaining wait events and totals them into a final number which also contributes to the overall elapsed time estimate of DB2 SQL Performance Analyzer. If table spaces and indexes are created with CLOSE YES option, they must often wait for VSAM to physically open the data sets which contain them. Opening data sets takes time, accounted for by DB2 SQL Performance Analyzer. Similarly, the bind process for dynamically issued SQL, such as SPUFI or QMF, and the impact of lock requests determined from the complete locking strategy is shown.

Elapsed Time

Wait time for the entire query, the Elapsed Time depicted in the cost summary block, is a function of the wait times for all physical synchronous read I/O, list and sequential prefetch I/O, asynchronous write I/O, binding, locking, opening and closing data sets and sorting, plus the CPU processing requirements of all steps in the access path.
Sample DB2 SQL Performance Analyzer detail trace report

Detail trace report

ANL3098I *** CONFIGURATION NOTE:
The overall rating for this system is totmips Million Instructions/Second
Each individual engine will operate at miprate Million Instructions/Second

ANL3036I *** The assumed Overhead for Connection Type of attach has been
estimated in Millions of Instructions for these BATCH SQL Stmts:
Thread Mgmt = thrdpl Attach = atchpl Appl = applpl

ANL3037I *** Estimated device DASD Service Times are:
Sync Read 4K Page = syncrd Prefetch block of 32 4K Pages = prefrd
Sync Read 8K Page = sync8b Prefetch block of 16 8K Pages = pref08
Sync Read 16K Page = sync16 Prefetch block of 8 16K Pages = pref16
Sync Read 32K Page = sync32 Prefetch block of 4 32K Pages = pref32
Async Wrt 4K Pages = asyn4k Async Write of 16K Page Block = asyn16
Async Wrt 8K Pages = asyn8k Async Write of 32K Page Block = asyn32

ROWS PROCESSED = norows PERCENT TABLE PROCESSED = pctrow
COLS PROCESSED = cols BOOLEAN FILTER FROM OPT = bolean
DATA PAGES READ= datapg INDEXED LEAF PAGES READ = leaves
SYNC READ I/O = syncio TABLE = tabrio INDEX = idxrio
PREFETCH I/O = seqprf TABLE = tabprf INDEX = idxprf
ASYNC WRT I/O = asywrft TABLE = tabasy INDEX = idxasy
GET PAGE CALLS = getpag SYSIO = sysio LOGIO = logio
SYNC READ = syncl Prefetch = prefpl Async WRT = asyp
GET PAGES = gtpgpl SYSTEMS = syspl LOG WRITE = logpl
DECOMPRESS= decode COMPRESS = encode HIPERPOOL = hpool
FETCH ROW = ftcpl LOCK/and so on = oddpl PROCESSES = procpl
PREDICATE = npred CLASS 1 = cls1pl OTHER O/H = othrpl

THE TOTAL CUMULATIVE PATH LENGTH FOR THIS QUERY IS cumpl M INS.
RESULTING IN A TOTAL CPU TIME OF cputim SECONDS CONSUMED OVERALL.
DB2 WILL PUT CLASS 1 CPU TIME OF cl1tim SECONDS IN SMF 101 RECORD.
DB2 SHOWS ADDITIONAL CPU TIME OF db2tim SECONDS IN SMF 100 RECORD.
ESTIMATED TOTAL LOGICAL I/O CALLS = logicio (EXCLUDING SYSTEM) AND
ESTIMATED TOTAL PHYSICAL I/O CALLS = physio WITH HIT RATIO = buffhit
WAIT TIME FOR SYNC READ I/O = wait1 PREFETCH I/O = wait2
WAIT TIME ON ASYNC WRITE I/O = wait3 TOTAL IWAIT = waitim
WAIT TIME FOR VSAM OPEN/CLOSE MACROS, BINDING AND LOCKING = waitd2

The following list is an explanation of all variables estimated in DB2 SQL
Performance Analyzer Detail Trace Report:

totmips
The total MIPS rating for the entire processor complex

miprate
The rated millions of instructions per second (MIPS) for a single processor engine
thrdpl  Path length associated with thread management, in millions of instructions
atchpl  The attach facility processor usage, in millions of instructions
applpl  Estimated application program path length, in millions of instructions
device  Type of disk device holding most/all of the database accessed
syncrd  Time to read one 4 KB page from the disk through a synchronous read I/O
sync08  Time to read one 8 KB page from the disk through a synchronous read I/O
sync16  Time to read one 16 KB page from the disk through a synchronous read I/O
sync32  Time to read one 32 KB page from the disk through a synchronous read I/O
prefrd  Time to read a block of “n” 4 KB pages from the disk through sequential or list prefetch I/O
pref08  Time to read a block of “n” 8 KB pages from the disk through sequential prefetch I/O
pref16  Time to read a block of “n” 16 KB pages from the disk through sequential prefetch I/O
pref32  Time to read a block of “n” 32 KB pages from the disk through sequential prefetch I/O
asyn4k  Time to write a block of 4 KB pages to disk through asynchronous write I/O
asyn8k  Time to write a block of 8 KB pages to disk through asynchronous write I/O
asyn16  Time to write a block of 16 KB pages to disk through asynchronous write I/O
asyn32  Time to write a block of 32 KB pages to disk through asynchronous write I/O
norows  Estimated number of rows returned by this query
cols   Estimated number of columns fetched by this query
pctrow  The percentage of rows processed from the table (always given)
bolean  The estimated Boolean factor for the query (only sometimes given)
datapg  Estimated number of data pages to be read from the table
leaves  Estimated number of index leaf pages to be read from the index
syncio  Total number of synchronous read I/Os issued for this step
tabrio  Portion of SYNCIO read from the table
idxrio  Portion of SYNCIO read from the index
seqprf  Total number of sequential and list prefetch I/Os issued for this step
tabprf  Portion of SEQPRF read from the table
idxprf  Portion of SEQPRF read from the index
asywrt  Total number of asynchronous writes issued for this step
tabasy  Portion of ASYWRT written to the table
idxasy  Portion of ASYWRT written to the index
getpag  Total number of Getpages estimated, including dynamic binds
sysio   Estimated system-oriented I/O, for catalog access, binding, and so on.
logio   Estimated log activity for writing updated records
procpl  Path length for most processes in access path, in millions of instructions
cls1pl  Path length attributable to class 1 activities, in millions of instructions
othrpl  Path length absorbed by DB2, in millions of instructions
syncpl  Path length to issue synchronous read I/Os, in millions of instructions
prefpl  Path length to issue sequential and list prefetch I/Os, in millions of instructions
asynpl  Path length to issue asynchronous write I/Os, in millions of instructions
gtpgpl  Path length to issue Getpage requests, in millions of instructions
syspl   Path length accounting for various system activities, in millions of instructions
logpl   Path length to issue log writes, in millions of instructions
ftchpl  Path length to fetch rows and columns, in millions of instructions
oddpl   Path length for application-specific activity, such as cursor manipulation
npred   Number of predicates identified by DB2 SQL Performance Analyzer in SQL statement
decode  Path length to extract rows, using hardware assist or software
encode  Path length to compress data rows before writing out to disk
hpool   Path length for Hiperpool accesses (reads and writes)
cls1pl  PROCPL + GTPGPL + SYNCPL + FTCHPL + SYSPL + ODDPL + DECODE + HPOOL (Reads), essentially all the class 1 processing
othrpl  PREFPL + LOGPL + ASYNPL + ENCODE + HPOOL (Writes) and THRDPL + ATCHPL + APPLPL on first block, representing all the processing absorbed by DB2 on your behalf
totpl   CLS1PL + OTHRPL, represented as a cumulative total
cump1  TOTPL, total path length as a cumulative total, except where multiple 
index and subquery are concerned, where this field reflects individual step 
totals

cputim  TOTPL / MIPRATE (also on cost summary report)

c1ltim  CLS1PL / MIPRATE

db2tim  OTHRPL / MIPRATE

logcio  SYNCIO + SEQPRF + ASYWRT + LOGIO + SORTIO + SQRYIO + JOINIO; 
essentially the count of all I/Os issued

buffhit  The percentage of I/O satisfied in the buffer pool, negating the need for 
physical I/O to disk

physio  LOCIO * (1 - BUFFHIT) (also on cost summary report)

wait1  Synchronous read I/O wait time

wait2  Sequential and list prefetch I/O wait time

wait3  Asynchronous write I/O wait time

waitd2  Other wait time for activities such as binding, locking, and open/close

waitc1  CL1TIM + DB2TIM, or wait time for class 1 activities

waitim  (WAIT1 + WAIT2 + WAIT3) * (1 - BUFFIT)

elapse  WAITIM + WAITC1 + WAITD2 (also on cost summary report)

qunits  A numeric value representation of the cost of the SQL statement, weighted 
towards the CPU processing cost, and including the I/O drive CPU time and 
DB2 CPU time (also on Cost Summary Report)

money  The final cost calculated, in monetary terms, based on parameters which 
price out the CPU, I/O, and elapsed times for the SQL statement (also on 
Cost Summary Report)

---

**QLIMIT (installation limits) report**

The QLIMIT report, which summarizes the entire run into a single record per 
query is available for SQL PA Batch and TSO users.

You can use the QLIMIT report for audit trail analysis. The QLIMIT report 
provides a summary to highlight irregular or inefficient queries.

The QLIMIT DD card allocates the report in batch runs while the Query Limits 
report on the SQL PA Output Data Sets panels do so under TSO. Sample JCL that 
shows the proper definition of this report is located in the *hiqual*.SANLJCL library, 
in members *ssid*JBUG, *ssid*JPA, and *ssid*JQST.

Use the output file for your analysis, because it provides a one line cost synopsis 
of each query evaluated with warning flags. The output file includes the SQL 
CODES for most statements that SQL PA was not able to evaluate (for example,
illegal use of parameter markers or table was not found). When DBRMs are processed, the member name where the statement belongs is populated. The QLIMIT report contains the following information:

5 Warning Flags
Set to Y or N to indicate whether each query has exceeded SQL PA limits for CPU time, elapsed time, physical I/O count, Qunits, or monetary costs, in that order. If there is an error during the processing of a query, the flags are set to E and you can interrogate the next field.

Lines that represent SQL statements and costs are marked with a hyphen (-) if they do not exceed the CPU, I/O, elapsed time, query units, or monetary costs, to make the Y stand out more prominently.

Return Code
Set to 0 if successful, or returns the error code encountered during query processing. This code corresponds to the typical SQL PA error message format: ANLxxxx, where xxxx is this four-digit return code.

Query Number
Value assigned by SQL PA during processing, to cross reference the query in the SQL PA Summary Cost, Explain, and Detail Trace reports.

Type
Identifies the statement type. The possible statement type values are shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>SELECT</td>
</tr>
<tr>
<td>IS</td>
<td>INSERT</td>
</tr>
<tr>
<td>UE</td>
<td>UPDATE</td>
</tr>
<tr>
<td>DE</td>
<td>DELETE</td>
</tr>
<tr>
<td>SD</td>
<td>SELECT with FOR UPDATE OF</td>
</tr>
<tr>
<td>CB</td>
<td>Correlated subselect or fullselect</td>
</tr>
<tr>
<td>NB</td>
<td>Non-correlated subselect or fullselect</td>
</tr>
<tr>
<td>TX</td>
<td>Table expression</td>
</tr>
<tr>
<td>TR</td>
<td>WHEN clause on CREATE TRIGGER</td>
</tr>
<tr>
<td>UN</td>
<td>UNION</td>
</tr>
<tr>
<td>UA</td>
<td>UNION ALL</td>
</tr>
</tbody>
</table>

CPU Time
DB2 SQL Performance Analyzer CPU time estimate for this query.

Elapsed Time
DB2 SQL Performance Analyzer elapsed time estimate for this query.

I/O Count
DB2 SQL Performance Analyzer physical I/O count estimate for query, after applying the buffer hit ratio.

QUNITs
DB2 SQL Performance Analyzer Qunits estimate for this query.

Monetary Cost
DB2 SQL Performance Analyzer Cost estimate for this query, in the local currency.
Member name
Name for the DBRM members.

Some users have automated their analysis of SQL PA output and might want to scan the QLIMIT output file through programs and take action on overruns. The file format is shown in the following figure.

```
<table>
<thead>
<tr>
<th>CLIM</th>
<th>ELIM</th>
<th>QLIM</th>
<th>MLIM</th>
<th>RETCODE</th>
<th>QUERYNO</th>
<th>TYPE</th>
<th>CPU TIME</th>
<th>ELAPSED</th>
<th>IOCOUNT</th>
<th>QUNITS</th>
<th>COSTS</th>
<th>MEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>A1</td>
<td>A1</td>
<td>A1</td>
<td>X 14</td>
<td>I10</td>
<td>I4</td>
<td>X 11,5</td>
<td>X 11,3</td>
<td>X 110</td>
<td>X 19</td>
<td>XF13,4</td>
<td>XA8</td>
</tr>
</tbody>
</table>
```

where A1 is 1 alphabetic character
X is a blank character
In is Integer of n Digits
Fn,m is Fixed Decimal of n Digits with m decimal places

Figure 55. Installation limits file format

A sample QLIMIT file is shown in the following figure.

```
10:05:35.998 SQL Performance Analyzer Version 4.1.0
05-23-2009 Query Limits Report Level 41-4100
CEIQ$ Error Queryno Type CPU Time Elapsed Phys I/O Qunits Monetary Cost Member Reason
----- 0 100005112 UE 0.14755 1.224 18 33 0.2039
----- 0 100012552 UE 0.01402 1.600 19 4 0.1964
----- 0 100012907 UE 0.01402 1.600 19 4 0.1964
----- 0 100005055 ST 0.01296 1.089 18 3 0.1848
----- 0 100005217 UR 0.00056 0.460 1 1 0.0114
----- 0 100007650 ST 0.00021 0.069 3 1 0.0302
----- 0 100009083 ST 0.00042 0.119 5 1 0.0504
----- 0 100009089 ST 0.00042 0.119 5 1 0.0504
----- 0 100009100 ST 0.00030 0.119 5 1 0.0504
----- 0 100009106 ST 0.00030 0.119 5 1 0.0504
----- 0 100009112 ST 0.00026 0.095 4 1 0.0403
----- 0 100009118 ST 0.00026 0.095 4 1 0.0403
----- 0 100009124 ST 0.00030 0.119 5 1 0.0504
----- 0 100009155 ST 0.00030 0.119 5 1 0.0504
----- 0 100009162 ST 0.00043 0.096 4 1 0.0403
----- 0 100009169 ST 0.00043 0.096 4 1 0.0403
----- 0 100009176 ST 0.00029 0.096 4 1 0.0403
----- 0 100009186 ST 0.00029 0.119 5 1 0.0504
----- 0 100009193 ST 0.00030 0.119 5 1 0.0504
----- 0 100009200 ST 0.00030 0.119 5 1 0.0504
----- 0 100009247 ST 0.00026 0.095 4 1 0.0403
----- 0 100009255 ST 0.00026 0.095 4 1 0.0403
----- 0 100009347 ST 0.00030 0.119 5 1 0.0504
```

Figure 56. Sample QLIMIT file

A header is provided on line 1 of the file to help remind users of the order of contents. SQL PA reports on the limits set during installation, which are used as warning flags for all DB2 SQL Performance Analyzer runs. The QLIMIT file is always created whenever the Explain or Detail Trace reports are requested in either TSO or batch modes.

You can sort the QLIMIT report from the ISPF interface by entering `SORT n A/D` where n is the column name and A or D indicates ascending or descending order. For example, `SORT 3 D` would sort the report in descending order based on the third column from the left, which is Queryno.
Predicate Analysis information

You can add a special predicate analysis report section to the Enhanced Explain and Detail Trace reports.

When operating with the parameter PRECISE ALL specified, SQL PA adds a special predicate analysis report section to both the Enhanced Explain and Detail Trace reports that are created during batch and TSO operations.

The predicate analysis is embedded like the performance notations of the SQL Advisor. The predicate analysis provides information about each predicate being processed, including those predicates generated by the DB2 optimizer for predicate transitive closure. It is no longer necessary to guess whether a predicate is Stage 1 or Stage 2 or what filter factor was calculated: the predicate analysis presents each predicate in a format to help you understand how the optimizer evaluated their predicates and deployed them for access path selection. Figure 57 on page 227 is an example of a Predicate Analysis report for a Cartesian join process.
In Figure 57, four predicates are ANDed together to access two tables from the catalog. But because there are no common join columns, this join is a Cartesian product join. You can see the format of each query in the PSUEDO TEXT, and the SQL PA ANALYSIS FOR QUERYNO 100000001

```
SELECT COUNT(*) FROM SYSIBM.SYSPACKSTM A, SYSIBM.SYSPACKDEP B
  WHERE A.LOCATION = ? AND
      B.DLOCATION = ? AND
      B.DCOLLID = ? AND
      A.COLLID <> ?
```

This is a Cartesian join, all rows to all rows, no join predicate.

... Queryno: 100000001
Predicate analysis
--------------

```
LEFT SIDE --> TABNO: 0 BLOCKNO: 0 PREDNO: 0
RIGHT SIDE --> TABNO: 0 BLOCKNO: 0 PREDNO: 5
PSUEDO TEXT:
  (((A.LOCATION=(EXPR) AND B.DLOCATION=(EXPR)) AND B.DCOLLID=(EXPR)) AND
   A.COLLID!=(EXPR))

LEFT SIDE --> LOCATION TABNO: 1 BLOCKNO: 0 PREDNO: 0
RIGHT SIDE --> VALUE TABNO: 0 BLOCKNO: 0 PREDNO: 0
PSUEDO TEXT:
  A.LOCATION=(EXPR)

LEFT SIDE --> DLOCATION TABNO: 2 BLOCKNO: 0 PREDNO: 0
RIGHT SIDE --> VALUE TABNO: 0 BLOCKNO: 0 PREDNO: 0
PSUEDO TEXT:
  B.DLOCATION=(EXPR)

LEFT SIDE --> DCOLLID TABNO: 2 BLOCKNO: 0 PREDNO: 0
RIGHT SIDE --> VALUE TABNO: 0 BLOCKNO: 0 PREDNO: 0
PSUEDO TEXT:
  B.DCOLLID=(EXPR)

LEFT SIDE --> COLLID TABNO: 1 BLOCKNO: 0 PREDNO: 0
RIGHT SIDE --> VALUE TABNO: 0 BLOCKNO: 0 PREDNO: 0
PSUEDO TEXT:
  A.COLLID!=(EXPR)
```

**Figure 57. Predicate analysis report: Cartesian join**

In Figure 57 four predicates are ANDed together to access two tables from the catalog. But because there are no common join columns, this join is a Cartesian product join. You can see the format of each query in the PSUEDO TEXT, and the
TYPE of predicate (in this case EQUAL), which largely determines its characteristics, such as STAGE 1 and BOOLEAN TERM. The computation of the FILTER factor shows the percentage of table rows that are filtered by this predicate and INDEX KEYFIELD, if Y, indicates that the column is part of an index key. FOR NEGATION, set to Y for Predicate 5, indicates that the predicate is a NOT (in this case <>). LITERALS HV indicates that host variables were used in the predicate and the left and right side, table numbers, and predicate numbers (AND or OR) provide additional information about each predicate in the statement.

Figure 58 provides another look at Predicate Analysis, for an OR of two Range predicates.

SQL PA ANALYSIS FOR QUERYNO 100000002

SELECT * FROM TDT690.L1000
  WHERE CIKEY < 10 OR CIKEY > 99999
...
QUERYNO: 100000002
PREDICATE ANALYSIS
------------------
QBLKNO: 1 PREDNO: 1 FILTER: 0.0199000 TYPE: OR JOIN PRED? N
ADDED BY PTC? N FOR NEGATION? N LITERALS:
LEFT SIDE --> TABNO: 0 BLOCKNO: 0 PREDNO: 2
RIGHT SIDE -> TABNO: 0 BLOCKNO: 0 PREDNO: 3
PSUEDO TEXT: (L1000.CIKEY<1 OR L1000.CIKEY>9)

QBLKNO: 1 PREDNO: 2 FILTER: 0.0100000 TYPE: RANGE JOIN PRED? N
ADDED BY PTC? N FOR NEGATION? N LITERALS: 1
LEFT SIDE --> CIKEY TABNO: 1 BLOCKNO: 0 PREDNO: 0
RIGHT SIDE --> VALUE TABNO: 0 BLOCKNO: 0 PREDNO: 0
PSUEDO TEXT: L1000.CIKEY<1

QBLKNO: 1 PREDNO: 3 FILTER: 0.0100000 TYPE: RANGE JOIN PRED? N
ADDED BY PTC? N FOR NEGATION? N LITERALS: 9
LEFT SIDE --> CIKEY TABNO: 1 BLOCKNO: 0 PREDNO: 0
RIGHT SIDE --> VALUE TABNO: 0 BLOCKNO: 0 PREDNO: 0
PSUEDO TEXT: L1000.CIKEY>9

Figure 58. Predicate analysis report: OR

The information in the Predicate Analysis report gives SQL PA users an advantage in understanding how each of the predicates are evaluated by DB2. The final composite filter factor for AND predicates is not currently computed (it is always 1.0) but is accurately assessed for OR predicates.

This information is automatically included as part of the Enhanced Explain and Detail Trace reports when PRECISE ALL is chosen.
Host Variables information

When a DBRM is processed as input, SQL Performance Analyzer extracts host variable information from the module. The extracted information displays in both the enhanced explain (ANLREP) and detail trace (QTRACE) reports, immediately following the Predicate Analysis report (if selected).

If there are no host variables, no report is provided for that SQL statement. The host variables are numbered in the order of their appearance in the statement. Key attributes, such as the data type, length, precision and scale (for decimal), input or output, along with the full name of each host variable are displayed. A sample Host Variables report is shown in the following figure.

```
QUERYNO: 100004702
HOST VARIABLES USED
-------------------
HV 1 DCLIDX.NAME
USAGE: OUTPUT DATA TYPE: 448 VAR CHARACTER CCSID: 0
DATA LENGTH: 12B (PRECISION: 0 SCALE: 0 PACKED DECIMAL ONLY)
HV 2 DCLIDX.CREATOR
USAGE: OUTPUT DATA TYPE: 448 VAR CHARACTER CCSID: 0
DATA LENGTH: 12B (PRECISION: 0 SCALE: 0 PACKED DECIMAL ONLY)
HV 3 DCLIDX.TBNAME
USAGE: OUTPUT DATA TYPE: 448 VAR CHARACTER CCSID: 0
DATA LENGTH: 12B (PRECISION: 0 SCALE: 0 PACKED DECIMAL ONLY)
HV 4 DCLIDX.TBCREATOR
USAGE: OUTPUT DATA TYPE: 448 VAR CHARACTER CCSID: 0
DATA LENGTH: 12B (PRECISION: 0 SCALE: 0 PACKED DECIMAL ONLY)
HV 5 DCLIDX.UNIQUERULE
USAGE: OUTPUT DATA TYPE: 452 CHARACTER CCSID: 0
DATA LENGTH: 1 (PRECISION: 0 SCALE: 0 PACKED DECIMAL ONLY)
HV 6 DCLIDX.COLCOUNT
USAGE: OUTPUT DATA TYPE: 500 SMALL INTEGER CCSID: 0
DATA LENGTH: 2 (PRECISION: 0 SCALE: 0 PACKED DECIMAL ONLY)
HV 7 DCLIDX.CLUSTERING
USAGE: OUTPUT DATA TYPE: 452 CHARACTER CCSID: 0
DATA LENGTH: 1 (PRECISION: 0 SCALE: 0 PACKED DECIMAL ONLY)
```

Figure 59. Host Variables report

The Easy Explain report

An Easy Explain report can consist of several pages.

The individual pages are numbered as x-y, where x denotes the report number and y is the page number within the report.

If the keyword used is QNO, PKG, or PLN, the query or statement number is shown at the top of the report. If the keyword is PLN or PKG, the DBRM and plan names followed by the actual SQL statement are also shown.
If the QMF keyword is used, the associated SQL statement is shown in the same form in which it was saved in QMF.

If the SQL keyword is used, the specified SQL statement is shown in the same format as you stated.

Each report lists the EXPLAIN-supplied information. This part of the report is a formatted listing of all the data taken from a row in the PLAN_TABLE table at the specific server location. For a description of data produced by the EXPLAIN function, see *DB2 UDB for OS/390 and z/OS V8 Administration Guide*.

The next section of the report shows the access path chosen by DB2.

If an index is used, that is, an access path other than a table space scan, index information is included in the report. This part of the report contains information about the following types of index-related data:
- Full key cardinality
- First key cardinality
- Key range
- Space allocated
- Number of pages used
- Number of levels in the index
- Size of subpage
- Uniqueness of the index
- Close and erase rules
- Buffer pool name
- Cluster ratio
- Clustering
- Prefetch factor
- I/O factor
- Index Type
- Piecesize

If an index is used, the next section of the report lists all the key columns in the index. If a matching index scan is used in the access path, the report indicates the columns that are used in the index scan. Used columns are indicated by an arrow (<=) in the Key Used column. The number of arrows corresponds to the contents of the matching columns field in the EXPLAIN data.

If LEVEL=KEYdist has been specified and RUNSTATS has produced key distribution information for the first column in the index in SYSIBM.SYSCOLDIST, a section in the report contains the distribution of the 10 (or fewer) most frequently used key values.

The next part of the report shows information about the accessed table and its corresponding table space. The information included in this part is:
- Name of table
- Number of rows and columns in the table
- Row length
- Name of Edit and Validation Procedures
- Name of table space and data base
- Name of default storage
- Creator of table and table space
- Number of tables in the table space
- Size of the page in the table space
- Number of active pages in the table space
If table space scan was selected as the access path method, Easy Explain prints a list of all available indexes on the subject table. This list includes the column names on which the indexes are built (a maximum of 12 columns, or the number of names that fit on the print line is listed).

If a plan name was specified using the PLN (or PLAN) keyword, or a query number was specified using the QNO (or QUERYNO) keyword for a miniplan created by the BIND process, only the last section of the report is shown. This section lists the following bind information:

- Bind date and time
- Isolation level
- Acquire and release points
- Plan validity
- Plan binder and owner
- Validation time
- Plan size
- Cache size
- Number of plan and system entries
- Number of SQL statements in plan
- Name of current server
- Degree of parallelism
- Disconnect option
- SQL rules

This part of the report is only produced with the first SQL statement of the plan.

If you are authorized to store and list plan information, the report shows the items in the previous list for the last four generations of the plan. DBRM and package information as outlined in the following list is also included in the report:

- DBRM/Package name
- Precompile date and time
- Source language
- Number of SQL statements
- Characterset
- Use of comma
- Use of Decimal(31)
- Source type (DBRM or Package)
- DB2 release when precompiled

After normal completion, the last page of the output shows a listing of all the input commands. Each command is prefixed with the report number. For each SQL, QUERYNO, and QMF statement, the access path is listed next to the input.
request. For each Plan or Package statement, the access path is listed for every SQL statement in the plan. To the right of the access path information, the report page number (x-y) is shown, unless the Level is Summary. In this way, you can use the summary page as a table of contents.

The top of the summary page contains a listing of the DB2 environment. This listing includes information such as:

- Level of Easy Explain, DB2, and MVS
- DB2 subsystem ID
- Location name
- MVS system ID
- Job name (if batch)
- Batch or TSO
- Start and stop time
- Run time

For each SQL statement in the plan or package, the summary page shows if the application module is a DBRM (D) or package (P) (see in the following figure), if a List Prefetch (L) or Sequential Prefetch (S) scan was used (see in the figure), and if a table space scan (*) or non-matching index scan (+) was selected by DB2 (see in the figure).

![Figure 60. Application module summary](image)

Topics:

- "Package reports"
- "Remote packages reports" on page 236
- "Easy Explain report review" on page 237
- "Easy Explain tabular report reference" on page 240

Package reports

There are a variety of package reports available.

When a plan is bound, you can include a package list that consists of wildcard characters, for example COLLID3.* or *. However, using wildcard characters can result in a plan pointing at thousands of packages with an even greater number of explainable SQL statements.

To prevent packages with SQL statements from being explained unnecessarily, Easy Explain looks for generic package lists in the plan being explained. If the collection ID or the package name in a package list was specified as an asterisk (*), Easy Explain determines the number of packages that belong to the package list specification.

If more than 100 packages belong to the package list specification, a report with all the package names is produced, but no SQL statements in these packages are
explained. The collection IDs are not listed. If any DBRMs belong to the plan, SQL statements in these DBRMs are explained.

The plan TESTPLAN was bound with a package list of *.*, a plan statement therefore results in the report shown in the following figure.

<table>
<thead>
<tr>
<th>Location: CPHMVS1_DB2X</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ANLI641</strong> Collection specification <em>.</em> has more than 100 packages</td>
</tr>
<tr>
<td>Use PACKAGE=collid.name(version) to get a detailed listing</td>
</tr>
</tbody>
</table>

**2005-04-30**  
*** Explained EXPLAIN Information ***  
Report 1  
Information for Plan: TESTPLAN  
*** Package Report ***  
Page 1-001  
==============================================================================

**Figure 61. TESTPLAN package report**

If more than 10 but less than 100 packages belong to the package list specification, a report with all the package names is produced, but no SQL statements in these packages are explained. This report shows the collection IDs, the creator and owner names, the version identifications, and the dates of precompilations. If any DBRMs belong to the plan, SQL statements in these DBRMs are explained.

The plan LARGPLAN was bound with a package list of E23PACK.*, a plan statement therefore results in the report shown in the following figure.
When a package is being explained, you might specify the collection ID, the package name, or both as generic names using an asterisk (*) as a wildcard character. For example, you might specify:

- Package=e23*.*
- Package=e23pack.*
- Package=e23pack.e23*
- Package=e23pack.e23main

If a wildcard character is used, Easy Explain counts the total number of SQL statements that belong to the packages (the total number of SQL statements also includes non-explainable statements).

If the number of packages is more than one and the total number of SQL statements in these packages is larger than 300, Easy Explain does not explain any SQL statements unless the FORCE=YES subparameter is specified. Instead the mini report shown in the following figure is produced.

**Figure 62. LARGPLAN package report**

When a package is being explained, you might specify the collection ID, the package name, or both as generic names using an asterisk (*) as a wildcard character. For example, you might specify:

- Package=e23*.*
- Package=e23pack.*
- Package=e23pack.e23*
- Package=e23pack.e23main

If a wildcard character is used, Easy Explain counts the total number of SQL statements that belong to the packages (the total number of SQL statements also includes non-explainable statements).

If the number of packages is more than one and the total number of SQL statements in these packages is larger than 300, Easy Explain does not explain any SQL statements unless the FORCE=YES subparameter is specified. Instead the mini report shown in the following figure is produced.

---

**2005-04-30  *** Explained EXPLAIN Information *** Report 1**

Information for Plan: LARGPLAN  *** Package Report *** Page 1-004

============================================================================
Location CPHMVS1_DB2X

ANL164I Collection specification E23PACK.* has more than 10 packages
  Use PACKAGE=collid.name(version) to get a detailed listing

<table>
<thead>
<tr>
<th>Package</th>
<th>Collection Id.</th>
<th>Creator</th>
<th>Owner</th>
<th>Exp</th>
<th>PC-Date</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>E23MAIN</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-12-20 1999-12-20-14.49.</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>E23MAIN</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-12-04 1999-12-04-19.16.</td>
<td></td>
</tr>
<tr>
<td>E23MAIN</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-12-03 1999-12-03-08.28.</td>
<td></td>
</tr>
<tr>
<td>E23MAIN</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-12-02 ANL420.5</td>
<td></td>
</tr>
<tr>
<td>E23MAIN</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-11-23 ANL420.4</td>
<td></td>
</tr>
<tr>
<td>E23MAIN</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-11-13 ANL420.2</td>
<td></td>
</tr>
<tr>
<td>E23PLAN</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-12-04</td>
<td></td>
</tr>
<tr>
<td>E23SERV</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-12-03 1999-12-03-08.12.</td>
<td></td>
</tr>
<tr>
<td>E23SERV</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-12-02 ANL420.5</td>
<td></td>
</tr>
<tr>
<td>E23SERV</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-11-25 ANL420.4</td>
<td></td>
</tr>
<tr>
<td>LONGVER</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1999-11-06 THIS_IS_A_VERY_LONGZERO</td>
<td></td>
</tr>
<tr>
<td>NEGZERO</td>
<td>E23PACK</td>
<td>DPGROTH</td>
<td>DPGROTH</td>
<td>Yes</td>
<td>1987-10-27</td>
<td></td>
</tr>
<tr>
<td>SQLUF1A0</td>
<td>E23PACK</td>
<td>DK11241</td>
<td>DK11241</td>
<td>No</td>
<td>0001-01-01</td>
<td></td>
</tr>
<tr>
<td>UFICMDS</td>
<td>E23PACK</td>
<td>ISTJE2</td>
<td>ISTJE2</td>
<td>No</td>
<td>0001-01-01</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 62. LARGPLAN package report**

When a package is being explained, you might specify the collection ID, the package name, or both as generic names using an asterisk (*) as a wildcard character. For example, you might specify:

- Package=e23*.*
- Package=e23pack.*
- Package=e23pack.e23*
- Package=e23pack.e23main

If a wildcard character is used, Easy Explain counts the total number of SQL statements that belong to the packages (the total number of SQL statements also includes non-explainable statements).

If the number of packages is more than one and the total number of SQL statements in these packages is larger than 300, Easy Explain does not explain any SQL statements unless the FORCE=YES subparameter is specified. Instead the mini report shown in the following figure is produced.
If the package specification does not contain any wildcard characters, the total number of SQL statements is less than 300, or only one package conforms to the specification, the SQL statements are explained. However, if the package exists in more than one version, only the most recent version is explained. In this case, a report showing all the versions for that package is produced. If you specify 

```
Package=e23pack.e23main(-7),gen=3
```

the report shown in the following figure is produced:

```
2005-04-30 *** Explained EXPLAIN Information *** Report 1
Page 1-001

Location CPHMVS1_DB2X

ANL372W The statement: PKG=E23PACK.E23*
will result in 25 packages
with a total of 2,042 SQL statements (including non-explainable).
If you want to explain that amount of packages, you should specify:
PKG=E23PACK.E23*,FORCE=Yes

Figure 63. Report where statements in packages is larger than 300
```

If the package specification does not contain any wildcard characters, the total number of SQL statements is less than 300, or only one package conforms to the specification, the SQL statements are explained. However, if the package exists in more than one version, only the most recent version is explained. In this case, a report showing all the versions for that package is produced. If you specify 

```
Package=e23pack.e23main(-7),gen=3
```

the report shown in the following figure is produced:

```
2005-04-30 *** Explained EXPLAIN Information *** Report 2
Information for Package: E23MAIN *** Version Report *** Page 2-000

Location CPHMVS1_DB2X

ANL166I Package: E23MAIN in Collection: E23PACK has the following versions

Pre-Comp'd Exp Gen Version Identification
------------------- ---- ---------------
1999-12-20 Yes 0 1999-12-20-14.49.03.726556
1999-12-19 Yes -1 1999-12-19-20.12.46.296787
1999-12-19 Yes -2 1999-12-19-16.31.06.369415
1999-12-19 Yes -3 1999-12-19-16.15.48.278743
1999-12-18 Yes -4 1999-12-18-09.17.18.591934
1999-12-17 Yes -5 1999-12-17-09.16.06.101113
1999-12-17 Yes -6 1999-12-17-08.44.59.541029
1999-12-17 Yes -7 1999-12-17-08.22.45.446834
1999-12-16 Yes -8 1999-12-16-12.04.03.828613
1999-12-15 Yes -9 1999-12-15-11.11.59.937500
1999-12-15 Yes -10 <version id not specified>
1999-12-09 Yes -11 1999-12-09-10.15.10.307455
1999-12-08 Yes -12 1999-12-08-11.10.46.584507
1999-12-08 Yes -13 1999-12-08-10.47.06.119000
1999-12-08 Yes -14 1999-12-08-10.31.08.242248
1999-12-05 Yes -15 1999-12-05-19.01.27.836759
1999-12-04 Yes -16 1999-12-04-19.16.42.852327
1999-12-03 Yes -17 1999-12-03-08.28.25.916853
1999-12-02 Yes -18 ANL420.5
1999-11-13 Yes -19 ANL420.4
1999-11-13 Yes -20 ANL420.2

Start version generation number specified: -7
Number of version generations requested: 3

Figure 64. Package version report
The listed report shows that the package exists in 21 versions, where the version that corresponds to generation number -7 was selected for explanation. The report further shows that you have asked for explanation of three generations (-7, -8, and -9). Instead of specifying a generation identification, the version identification could be specified. You can either specify the version ID in full or in combination with a wildcard character, as shown in the following example:

```
Package=e23pack.e23main.(1999-12-17-08.22.*), gen=3
```

A maximum of 100 generations are listed.

**Remote packages reports**

An application program can consist of DBRMs and packages. The packages are located on the same DB2 subsystem as the plan, but the packages can also be distributed to one or more remote locations.

The following figure shows an application plan named XYZ located at Location_A, which is the home server location. The application plan consists of two DBRMs plus a package list. This package list contains pointers to local and remote packages.

In the example, the package list points to the remote package Package_A, which is located at server Location_C. Package_B is a local package, and Package_C and Package_D are remote packages located at Location_B.

To explain all the static SQL statements for plan XYZ, Easy Explain automatically connects to all the locations where the plan has remote packages if the locations were specified at Bind time. In this way, Easy Explain reports explain data and catalog data for all static SQL statements in the plan XYZ.

For example, you might want only to explain Package_D at Location_B. By using the location subparameter on the Package statement, you can instruct Easy Explain to explain that package at the specified location.
All the explain reports and package reports show the location name for the individual SQL statements and packages.

**Easy Explain report review**

If an application does not achieve the expected performance, investigate the individual Easy Explain report pages carefully.

Performance analysis is a complex task. It is important to know if the environment where you are running Easy Explain for data analysis is the production environment, or a test environment. The main performance factors to review include the following items, but other factors might be the cause for your performance issues:

**Access Path Chosen**

Typically, avoid table space scan and non-matching index scans, unless you intend to access all rows in a given table. If one or more indexes exist for the table, try to reconstruct the SQL statement so that a better access path is chosen by DB2. Alternatively, you could consider creating an index.

Also, try to avoid internal sorts. If you are joining, avoid a merge scan join because a sort is involved.
**Clustering versus Clustered**
If a clustering index was selected by DB2, the Easy Explain report shows if the actual index also is clustered and the clustering ratio is shown. If the clustered value is N (or the cluster ratio is less than 95%), the table space might need reorganization to bring the data rows into clustering sequence again.

**Number of matching columns**
If a matching index scan is selected by DB2, verify that the number of columns used in the index is what you expect. Locate the <=== arrow under the Key Used heading in the index section. Every column used in the index search is marked with <===. The access path box and the Summary Page show the ratio between number of matching columns and number of key columns in the index.

**Tables per tablespace**
In the table space section, locate the heading Tabls/TS. This value gives the number of tables in the table space. Normally, you would expect to have only one table in the table space.

**Active pages versus pages with rows**
If you are doing table space scans, verify that the number of pages with rows (heading Pages w/Rows in the table section) is about the same size as the value under Pages Active. The value under heading PctPages in the table section needs to be as close to 100% as possible.

If the table space scan takes place for a non-segmented table, and more than 100 pages with no table rows are scanned, and this number of pages is more than 10% of the total table space pages, the number of pages is shown in the access path box.

The following example is the first page of an Easy Explain report where you are authorized to store and list plan information (PLANINFO=YES).
After explaining an application using Easy Explain where the access path and other factors are satisfactory, save the data for historical purposes. The historical analysis data gives you a base for comparing past and present performance. Save the Easy Explain report together with the application documentation.

The report contains columns with plan data from the last four plan generations. If there is a discrepancy in the column data within a row, a marker (<<<) is shown in the rightmost column.

Look for changes in the number of SQL statements, validate, isolation, acquire, and release. A change in any of these areas might influence the performance of the application plan.

The bottom part of the subreport shows data for each DBRM and package in the most recent plan generation. The report states the precompile date and time, the programming language, the number of SQL statements, single or double byte
character set, use of comma, use of decimal(31), type of source, and DB2 release indicating when the module was precompiled.

Often an inconsistent definition of host variables compared to the corresponding column definition results in an inefficient access path selection.

In the example shown in the following figure, the access path selected is Table Space Scan even though an index is defined on the only column referenced in the WHERE-clause. By specifying ...HOSTVAR=Yes,INDEX=T,... on the plan or package statement, Easy Explain shows both the index and the host variable definitions. As the example shows, DB2 has selected Table Space Scan because the column definition is 3 characters (see 1), but the corresponding host variable is defined as 4 characters (see 2). By changing the host variable definition to 3 characters, a matching index scan is selected by the DB2 Optimizer.

```
DECLARE C1 CURSOR WITH HOLD FOR
SELECT DEPTNO, DEPTNAME, LOCATION
FROM DSN8610.DEPT
WHERE DEPTNO = :HOSTVAR_STRUCTURE.DEPARTMENT_NUMBER
ORDER BY DEPTNO

+------------------------------------------------------------------+
| Table space scan - no index will be used |
| Standard sequential PREFETCH will be performed |
| Lock mode is Share Lock for the page |
+------------------------------------------------------------------+

Key | No. Column Name | Col.Type | Lng | Null | Card. Order | Low2key | High2key | Used |
--- | --------------- | -------- | --- | ---- | ----------- | ------- | -------- | ---- |
1 DEPTNO           CHAR   3    No    14    Asc.   c'801       c'122

Host var. type | Length | Ind. Host variable name | (DB2 rel. 6.1 ) |
--------------- | ------ | ------------------------ | -----------------|
Fixed character | 4      | HOSTVAR_STRUCTURE.DEPARTMENT_NUMBER |
```

Figure 67. DBRM and Package generation data

Note: If a host variable has a related indicator variable, the Ind. column shows a Yes for the host variable, and the following line in the host variable report shows the indicator variable.

**Easy Explain tabular report reference**

The Easy Explain report is available in a shortened format that can be specified by setting LEVEL=TABular.

**Columns**

If LEVEL=TABular has been specified, the report shows information about the accessed table and its corresponding table space. The information includes the following columns:
• Query number
• Block number
• Plan Number
• Creator
• Creator of table and table space
• Table name
• Access type
• Matching columns
• Multiple index operation sequence
• Index only
• Sort unique
• Sort join
• Order by
• Group by
• Merge
• Tablespace lock mode
• Prefetch
• Column function evaluation
• Degree of access
• Access parallel group ID
• Join degree
• Join parallel group ID
• Parallel group ID for the parallel sort of the composite table
• Parallel group ID for the parallel sort of the new table
• Parallelism mode
• Page range
• When optimize
• Query block type
• Primary access type
• Table type

Sample
12009-08-24 *** Explained EXPLAIN Information *** Report 1
1 PKxxxx 4.1.00
FILE= userid.SQL.INPUT(TSTINPUT) Page 1-001
SQL Statement for Query No.100000001
=======================================================================
SELECT *
FROM ANLUSER0.EEEPLAN
WHERE NAME = 'FINDNAME'
=======================================================================
Breakdown of EXPLAIN information for Query No.100000001
=======================================================================
0PLAN_TABLE Data:
Blk Pln A Mat MIX IPPDCPTW TS SORTC SORTN Access Join
No. No. T Col Seq ORRFEMTO Lck UJOGM PID UJOGM PID Deg PID Deg PID
--- --- -- --- --- -------- --- ----- --- ----- --- --- --- --- ---
1 1 I 0 0 N D T IS NNNN - NNNN - - - - -
Type=SELECT
IX=JOEUSER.X_E3PLN1
Col UCSCSH Clu First Full Avg
Cnt RGDPA Nleaf Nlvls Rat Keycard Keycard Keylen Statstime
--- ----- ----- ----- ----- ----- ----- ----- --- -------------------
2 PNYNN 32 2 96 63 2252 21 2010-05-13-11.08.50
### TB=ANLUSER0.EEEPLAN

<table>
<thead>
<tr>
<th>Rows</th>
<th>Npages</th>
<th>Max</th>
<th>Avg</th>
<th>Count</th>
<th>Statetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0001-01-01-00.00.00</td>
</tr>
</tbody>
</table>

### TS=JOEUSER.ANLSPACE

<table>
<thead>
<tr>
<th>Partitions</th>
<th>Nactive</th>
<th>Statetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0001-01-01-00.00.00</td>
</tr>
</tbody>
</table>

### Key

<table>
<thead>
<tr>
<th>No.</th>
<th>Column Name</th>
<th>Col.Type</th>
<th>Lng</th>
<th>Null</th>
<th>Card.</th>
<th>OPGMH</th>
<th>Low2key</th>
<th>High2key</th>
<th>Usd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NAME</td>
<td>VARCHAR</td>
<td>128</td>
<td>No</td>
<td>169</td>
<td>A</td>
<td>c'AASFS</td>
<td>c'XSR_REGI&lt;==</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PARM_COUNT</td>
<td>SMALLINT</td>
<td>2</td>
<td>No</td>
<td>21</td>
<td>A</td>
<td>x'8001</td>
<td>x'8029</td>
<td></td>
</tr>
</tbody>
</table>

Information for ALL available indexes is requested by the user:

**IX=JOEUSER.X_E3PLN1**

**IX=JOEUSER.X_E3PLN2**

### Information for ALL available indexes is requested by the user:

#### IX=JOEUSER.X_E3PLN1

#### IX=JOEUSER.X_E3PLN2

### Key

<table>
<thead>
<tr>
<th>No.</th>
<th>Column Name</th>
<th>Col.Type</th>
<th>Lng</th>
<th>Null</th>
<th>Card.</th>
<th>OPGMH</th>
<th>Low2key</th>
<th>High2key</th>
<th>Usd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NAME</td>
<td>CHAR</td>
<td>8</td>
<td>No</td>
<td>-1</td>
<td>Asc</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CREATOR</td>
<td>CHAR</td>
<td>8</td>
<td>No</td>
<td>-1</td>
<td>Asc</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>BINDDATE</td>
<td>CHAR</td>
<td>6</td>
<td>No</td>
<td>-1</td>
<td>Asc</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>BINDTIME</td>
<td>CHAR</td>
<td>8</td>
<td>No</td>
<td>-1</td>
<td>Asc</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

1 SQL Performance Analyzer Plan Table Report Summary Page

<table>
<thead>
<tr>
<th>ANL</th>
<th>Release: 4.2.0 of 2013-10-23 APAR: PM97648</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODB2</td>
<td>Release: 8.1.5 DB2 Macro Change Level: V9R1M0</td>
</tr>
<tr>
<td>MVS</td>
<td>Release: 7.1.1 z/OS System id: SY4A</td>
</tr>
<tr>
<td>ODB2</td>
<td>System id: DSN9 Product Code: 5740-XYR</td>
</tr>
<tr>
<td>User Auth. id</td>
<td>JOEUSER TSO</td>
</tr>
<tr>
<td>Curr. SQL id</td>
<td>JOEUSER Plan Name: ANLEEE</td>
</tr>
<tr>
<td>Location id</td>
<td>DSNB</td>
</tr>
<tr>
<td>Coded Char. Set Id</td>
<td>37</td>
</tr>
</tbody>
</table>

Start date: 2009-08-24 Start time: 10:22:49
Stop date: 2009-08-24 Stop time: 10:22:49
Elapse time: 00:00:00

Default Level=TABULAR, Index=Y, Acctype=A, HostVar=Y
Options: Format=Y, PlnInfo=Y, SQLPA=Y, SQLS=079
Options: DSNBEXP=N
SET CURRENT DEGREE = '1'
Current Degree set to '1'
SET CURRENT SQLID = 'JOEUSER'
Current Authorization Id. specified
Part 4. Troubleshooting

The topics in this section provide you with information on administering the product.
Chapter 7. DB2 SQL Performance Analyzer troubleshooting

Use these topics to troubleshoot DB2 SQL Performance Analyzer.

Gathering diagnostic information

Before you report a problem with DB2 SQL Performance Analyzer to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all DB2 SQL Performance Analyzer problems:

- A clear description of the problem and the steps that are required to re-create the problem
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of DB2 that you are using and the type and version of the operating system that you are using

Provide additional information based on the type of problem that you experienced:

For online abends, provide the following information:

- A screen shot of the panel that you were using when the abend occurred
- The job log from the TSO session that encountered the abend
- The job log from the server
- A description of the task that you were doing before the abend occurred

For errors in batch processing, provide the following information:

- The complete job log
- Print output
- Contents of the any data sets that were used during the processing

Resetting the DB2 SQL Performance Analyzer REGISTRY

DB2 SQL Performance Analyzer includes a cleanup mechanism that resets all the SQL PA entries in the REGISTRY to their original status.

About this task

Occasionally, an entry might have an invalid status of in use in the DB2 SQL Performance Analyzer REGISTRY. The DB2 SQL Performance Analyzer REGISTRY table can be reset if entries are incorrectly marked as in use.

The DB2 SQL Performance Analyzer REGISTRY table contains an entry for each DB2 SQL Performance Analyzer generic plan table, holding an DB2 SQL Performance Analyzer secondary authorization ID, a place for the primary authorization ID of the user (set to ANL99999), an in use flag (Y or N) to help avoid conflicts, and a timestamp indicating last use.
By registering a PLAN_TABLE as in use, DB2 SQL Performance Analyzer gives exclusive control of that table to a specific user, for a portion of the time DB2 SQL Performance Analyzer is processing. The table is released as soon as it is no longer required, before the end of the DB2 SQL Performance Analyzer process.

**Tip:** To clean up any fragmented or frozen entries in the SQL PA REGISTRY, consider adding ssidRSET to your job scheduler so that it runs periodically.

Rarely, users might receive the message:

```
ANL1008E There are no secondary authids unused at this time.
Try again later.
```

DB2 SQL Performance Analyzer contains code to detect when the REGISTRY is full, and automatically resets any entries that are more than one hour old.

**Procedure**

To run the cleanup mechanism:

1. Locate and run hiqual.SANLSQL, member ssidRSET. ssidRSET is designed to run under TSO SPUFI, but you can also run it from a batch job.
2. Add ssidRSET to your job scheduler so that it runs periodically.

---

**TSO interfacing notes for installers**

The following topics contain special TSO interfacing notes for installers.

Topics:

- “DB2 SQL Performance Analyzer variables in the ISPF user profile”
- “Host variable support from any TSO input medium”
- “SQL PA supports upper and lower case”

**DB2 SQL Performance Analyzer variables in the ISPF user profile**

DB2 SQL Performance Analyzer makes liberal use of ISPF profile variable storage, and keeps the latest values for the TSO interface parameters there. Therefore, the latest values are retained between runs.

**Host variable support from any TSO input medium**

DB2 SQL Performance Analyzer allows you to specify host variable names in SQL submitted for analysis under TSO (and QMF), in addition to their typical use within application programs, and subsequent DBRMs.

Under normal circumstances, an EXPLAIN of an SQL statement under TSO or QMF would not allow use of application host variables (:Host), preferring the parameter marker (?) in its place. SQL PA adjusts automatically, so that if you want to isolate some SQL from a COBOL program, for example, SQL PA explains it and performs a cost analysis, replacing the :Host variables with the parameter marker automatically as it processes.

**SQL PA supports upper and lower case**

DB2 SQL Performance Analyzer handles SQL in both upper and lower case. All lower case values get translated to upper case before processing unless the
NLSCODE parameter is in effect (DBCS or Mixed Data names). SQL PA’s translation to upper case avoids any problems that might occur because of lower case DB2 object names.

**Applying user indexes to tables**

To improve Easy Explain performance, apply user indexes to three system catalog tables.

**Procedure**

Create the indexes shown in the following table in ASC order.

*Table 18. System catalog table indexes*

<table>
<thead>
<tr>
<th>Index Name</th>
<th>System Catalog</th>
<th>Table Column names</th>
</tr>
</thead>
<tbody>
<tr>
<td>XUSRDBRM</td>
<td>SYSIBM.SYSDBRM</td>
<td>(PLNAME, NAME)</td>
</tr>
<tr>
<td>XUSRFLDS</td>
<td>SYSIBM.SYSFIELDS</td>
<td>(TBCREATOR, TBNAME, NAME, FLDPROC)</td>
</tr>
<tr>
<td>XUSRSTMT</td>
<td>SYSIBM.SYSSTMT</td>
<td>(PLCREATOR, PLNAME, NAME, STMTNO, SEQNO)</td>
</tr>
</tbody>
</table>
Chapter 8. Tools Customizer troubleshooting

Use this information to diagnose and correct problems that you experience with Tools Customizer.

Gathering diagnostic information

Before you report a problem with Tools Customizer to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all Tools Customizer problems:

- A clear description of the problem and the steps that are required to re-create the problem
- Relevant screen captures
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of DB2 that you are using and the type and version of the operating system that you are using
- The Tools Customizer trace data set
- The Tools Customizer data store data set and the high_level_qualifier.SCCQTENU data set

Determining the trace data set name

You will need to identify the name of the trace data set if you cannot allocate the trace data set, the trace data set runs out of space, or IBM Software Support asks for it.

The name of the trace data set depends on the prefix setting in the TSO profile. To identify the name of the trace data set, you must know the prefix setting.

- If PREFIX is set, the name of the trace data set is prefix.CCQ.TRACE, where prefix is the TSO prefix that you specified in the profile.
- If NOPREFIX is set, the name of the trace data set is user_ID.CCQ.TRACE, where user_ID is your TSO user ID.
Chapter 9. How to look up message explanations

You can use several methods to search for messages and codes.

Searching an information center

In the search box that is located in the top left toolbar of any Eclipse help system, such as the IBM Information Management Software for z/OS Solutions Information Center, enter the number of the message that you want to locate. For example, you can enter DFS1065A in the search field.

Use the following tips to improve your message searches:

- You can search for information on codes by entering the code; for example, enter -327.
- Enter the complete or partial message number. You can use the asterisk wildcard character (*) to represent multiple characters, and you can use the question mark wildcard character (?) to represent a single character.

The information center contains the latest message information for all of the Information Management products that are included in the information center.

Searching for messages on the web

You can use any of the popular search engines that are available on the web to search for message explanations. When you type the specific message number or code into the search engine, you are presented with links to the message information in IBM information centers.
Part 5. Reference

The topics in this section provide you with technical references for the product.
Chapter 10. Reference information

This section contains reference information for SQL PA.

Topics:

- “ANLCNTL configuration parameters” on page 262
- “ANLPARM user parameters” on page 267
- “PLAN_TABLE contents” on page 284

Tools Customizer reference

Before you use Tools Customizer, you should understand the Tools Customizer terminology and the data sets that Tools Customizer uses during customization.

Tools Customizer terminology and data sets

Before you use Tools Customizer, you should understand the Tools Customizer terminology and the data sets that Tools Customizer uses during customization.

Tools Customizer terminology

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Products and components

How an IBM Tool is packaged determines whether it is referred to as a product or as a component in the Tools Customizer documentation and interface. An IBM Tool that is ordered as a stand-alone entity (that is, not as part of a solution pack) is referred to as a product. An IBM Tool that is part of a solution pack is referred to as a component. Some IBM Tools are available in both formats; therefore, the same IBM Tool can be referred to as a product or as a component depending on how it is packaged.

DB2 entry

You can customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool on one or more DB2 entries. A DB2 entry can be any of the following items:

DB2 subsystem

A distinct instance of a relational database management system (RDBMS) that is not part of a data sharing group. An example of a DB2 subsystem name is DB01.

DB2 group attach name

The name that is used by the TSO/batch attachment, the call attachment facility (CAF), DL/I batch, utilities, and the Resource Recovery Services attachment facility (RRSAF) as a generic attachment name. An example of a group attach name is DSG1.

DB2 data sharing member

A DB2 subsystem that is assigned by the cross-system coupling facility (XCF) to a data sharing group. An example of a DB2 data sharing member name is DB02.

Tools Customizer maintains the following lists of DB2 entries:

Associated list
The list of DB2 entries that are associated with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool. If the product to be customized requires DB2 entries, you can customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool only on DB2 entries that are in the associated list. When you customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, this list is displayed in the DB2 Entries, Associations, and Parameter Status section of the Customizer Workplace panel.

You can add and copy DB2 entries to the associated list. When you add or copy DB2 entries to the associated list, the entries are associated with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

Master list

The list of all DB2 entries that are defined but are not associated with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool. Tools Customizer obtains information about these DB2 entries either from entries that were created manually or from the customizations of other products that were discovered. If you remove a DB2 entry from the associated list, the DB2 entry is added to the master list. When you create a new DB2 entry, it is added to the master list, and when you associate the new entry with DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, it is removed from the master list and added to the associated list. The master list is displayed on the Associate a DB2 Entry for Product panel.

If the associated list does not have the DB2 entries on which you want to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, you can associate existing entries from the master list to the associated list.

You can create new DB2 entries and copy existing entries to the master list.

High-level qualifier

The high-level qualifier is considered to be all of the qualifiers except the lowest level qualifier. A high-level qualifier includes a mid-level qualifier.

Product parameters

Parameters that are specific to DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool. These parameters are defined by DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool and are stored in a data member that is defined by DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

LPAR parameters

Parameters on the local LPAR that are required to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool. These parameters are defined by Tools Customizer and are stored in an LPAR parameter data member.

DB2 parameters

Parameters for a DB2 entry. These parameters are defined by Tools Customizer and are stored in a DB2 parameter data member.

Status type

Product, LPAR, and DB2 entry status type
After you specify the product that you want to customize, the product, the LPAR, and the DB2 entries have a status. The status is partly based on whether required parameters are defined. For some products, LPAR parameters or DB2 parameters might not be required. In these cases, the status is Not Required.

To customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, all of the required parameters must be defined.

If required parameters for the product parameters, LPAR parameters, or DB2 parameters are not defined, the status of the parameters is Incomplete. Define values for parameters by manually editing them or by generating the customization jobs and specifying values for all of the required parameters that are displayed on the panels.

When values for all of the required parameters are defined, the status is Ready to Customize. Customization jobs can be generated only when all of the required parameters are defined and the status is Ready to Customize or Customized for the product parameters, LPAR parameters, and DB2 parameters for the DB2 entries on which DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool will be customized.

The following table shows the meaning of the status types. Each status is defined differently for each type of parameter.

<table>
<thead>
<tr>
<th>Status</th>
<th>Product</th>
<th>LPAR</th>
<th>DB2 entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete</td>
<td>The required product parameters are not defined, or the required product parameters are defined but LPAR parameters, DB2 parameters, or both are not defined.</td>
<td>The required parameters are not defined.</td>
<td>The required parameters are not defined.</td>
</tr>
<tr>
<td>Discovered</td>
<td>The product parameter definitions were discovered by using the product Discover EXEC.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ready to Customize</td>
<td>The required product, LPAR, and DB2 parameters are defined, the status is Ready to Customize or Customized for the LPAR and at least one associated DB2 entry. You can generate the customization jobs.</td>
<td>The required LPAR parameters are defined or LPAR parameters are not required.</td>
<td>The required DB2 parameters are defined or DB2 parameters are not required.</td>
</tr>
</tbody>
</table>
### Table 19. Status types for the product, the LPAR, and the DB2 entries (continued)

<table>
<thead>
<tr>
<th>Status</th>
<th>Product</th>
<th>LPAR</th>
<th>DB2 entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customized</td>
<td>The jobs are customized on the local LPAR.</td>
<td>The jobs are customized for the product or for all of the associated DB2 entries on the local LPAR.</td>
<td>The jobs are customized for the DB2 entry.</td>
</tr>
<tr>
<td>Errors in Customization</td>
<td>N/A</td>
<td>N/A</td>
<td>Errors occurred while the customization jobs were being generated.</td>
</tr>
<tr>
<td>Not Required</td>
<td>N/A</td>
<td>LPAR parameters are not required.</td>
<td>DB2 parameters are not required.</td>
</tr>
</tbody>
</table>

### Data sets that Tools Customizer uses during customization

Tools Customizer uses several unique data sets during the customization process. Familiarize yourself with these data sets before you begin to use Tools Customizer.

Several different data sets are required to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool with Tools Customizer. These data sets are supplied by DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, supplied by Tools Customizer, or allocated by Tools Customizer.

DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool provides the following data sets:

**Metadata library**
Contains the metadata for the product to be customized. Tools Customizer uses the metadata to determine which tasks, steps, and parameters to display on the Product Parameters panel, the LPAR Parameters panel, and the DB2 Parameters panel. This data set also contains the templates that Tools Customizer uses to generate the customization jobs.

The metadata library naming convention is `high_level_qualifier.SANLDYXIMZDENU`, where `high_level_qualifier` is all of the segments of the data set name except the lowest-level qualifier.

You specify the metadata library on the Specify the Metadata Library panel. READ access to this data set is required.

**Discover EXEC library (CLIST library)**
Contains the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC. When you customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, you can use the Discover EXEC to automatically retrieve and store product information, such as parameter values from an already customized product. Tools Customizer saves the discovered information in the data store.

The default name of the data set is the high-level qualifier for the metadata library plus a lowest-level qualifier. For DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, the lowest-level qualifier is `SANLDYXIMZCLSTDENU`. You can change the default value on the Discover Customized Product Information panel. EXECUTE access to this data set is required.
Tools Customizer provides the following data sets:

**Tools Customizer metadata library**
Contains the metadata for the DB2 and LPAR parameters that are required to customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool. Tools Customizer uses the metadata to determine which parameters to display on the DB2 Parameters panel and the LPAR Parameters panel. In addition, Tools Customizer uses information in the metadata library to determine whether additional DB2 and LPAR parameters need to be displayed on these panels. As you customize different products, different DB2 and LPAR parameters might need to be defined.

The default name of the data set is DB2TOOL.CCQ110.SCCQDENU. You can change the default value on the Tools Customizer Settings panel. READ access to this data set is required.

**Tools Customizer table library**
Stores information about jobs that are customized. Job information that is stored includes a description of the job, its member name and template name, the SSID, group attach name, and when the job was generated.

The default name of the data set is DB2TOOL.CCQ110.SCCQTENU. WRITE access to this data set is required.

Tools Customizer requires that the following data sets exist during the customization process. If the data sets do not exist, Tools Customizer automatically allocates them.

**Discover output data set**
Contains the output that is generated when you run the DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC. The DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool Discover EXEC retrieves the metadata and values for the parameters from a previous customization of DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

The default name of the data set is DB2TOOL.CCQ110.DISCOVER. You can change the default value on the Tools Customizer Settings panel or the Discover Customized Product Information panel. WRITE access to this data set is required.

**Data store data set**
Contains product, LPAR, and DB2 parameter values, and DB2 entry associations. Tools Customizer uses this data set to permanently store all information that is acquired about the product, DB2 subsystems or data sharing groups, and LPAR when you customize products on the local LPAR.

The default name of the data set is DB2TOOL.CCQ110.DASTOR. You can change the default value on the Tools Customizer Settings panel. WRITE access to this data set is required.

**Customization library**
Contains the customization jobs that Tools Customizer generates for DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool.

Tools Customizer checks whether a customization library name was specified for more than one instance of the same version of the same product. If the same customization library name is specified for more than one product of the same version, the CCQD123E message is issued to
prevent you from overwriting previously generated customization jobs. Ensure that you specify unique qualifier for the customization library for each instance of the product.

To customize DB2 SQL Performance AnalyzerDB2 Autonomics DirectorIMz Sample Tool, submit the members of the data set in the order in which they are displayed on the Finish Product Customization panel.

The data set naming convention is $hlq$.LPAR_name$.xyzvrm$, where:

- $hlq$ is the value of the Customization library qualifier field on the Tools Customizer Settings panel (CCQPSET)
- $LPAR_name$ is the four-character LPAR name
- $xyzvrm$ is the three-letter product identifier with the version, release, and modification level

For example, the data set name might be

DB2TOOL.PRODUCT.CUST.$MVS1$.XYZ410.

WRITE access to this data set is required.

Tools Customizer allocates the data sets for the discover output, the data store, and the customization library with the attributes that are shown in the following table:

<table>
<thead>
<tr>
<th>Data set</th>
<th>Organization</th>
<th>Record format</th>
<th>Record length</th>
<th>Block size</th>
<th>Data set name type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover output data set</td>
<td>PO</td>
<td>Variable block</td>
<td>16383</td>
<td>32760</td>
<td>LIBRARY</td>
</tr>
<tr>
<td>Data store data set</td>
<td>PO</td>
<td>Variable block</td>
<td>16383</td>
<td>32760</td>
<td>LIBRARY</td>
</tr>
<tr>
<td>Product customization library</td>
<td>PO</td>
<td>Fixed block</td>
<td>80</td>
<td>32720</td>
<td>LIBRARY</td>
</tr>
</tbody>
</table>

Restrictions:

- Multiple users cannot simultaneously share the discover output data set, data store data set, Tools Customizer metadata library, and metadata library.
- You cannot share the data store data set across multiple LPARs with shared DASD or copy the data store data set to another LPAR. Tools Customizer creates many cross-references between product and DB2 associations. Therefore, if you share or copy the data store data set, member names that are empty or that do not exist might be generated.

**ANLCNTL configuration parameters**

The following information describes the ANLCNTL configuration parameters.

The ANLCNTL parameters are not normally modified by DB2 SQL Performance Analyzer users.
The following parameters define a target host configuration. You can have multiple configurations (PDS members) in batch using the ANLCNTL DD statement to point to the hiqual.SANLDATA PDS member, and in TSO by having multiple members in the hiqual.SANLPARM PDS. For each member (subsystem, LPAR, or configuration), configure the following parameters of ANLCNTL:

**ANLKEYS nnnnn**
This control parameter is used to modify the DB2 SQL Performance Analyzer cost analysis. A value of 10000 tells DB2 SQL Performance Analyzer to use only Class 1 CPU times in its estimates, good for validation of DB2 SQL Performance Analyzer against an online monitor. A value of 02000 tells DB2 SQL Performance Analyzer to eliminate all processor usage from the calculations, ideal for building capacity planning models that roll up n times. These are combined, as 12000, to gain both effects at once. A value of 01000 tells DB2 SQL Performance Analyzer to process the parser only, and is helpful in isolating an SQL parsing problem if requested by IBM Software Support.

**Warning:** Do not attempt to run SQL PA with any other values in this parameter. There are additional, undocumented settings which do not give accurate results to the uninformed user.

**AUTHIDO**
The DB2 user ID is used to customize SQL PA. This ID owns the SQL PA objects that are created for the customization and must have SYSADM authority. The default value is ANLUSER0.

**BUFF08K nnnnnnnn**
Set the BUFF08K system parameter to the total number of 8 KB buffers in the 8 KB buffer pool of the dominant application. This value is used to set the number of pages per block for sequential prefetch and other I/O calculations. Use the most prominent 8 KB buffer pool here, even if several are used.

The default value is 00000500 buffers.

**BUFF16K nnnnnnnn**
Set the BUFF16K system parameter to the total number of 16 KB buffers in the 16 KB buffer pool of the dominant application. This value is used to set the number of pages per block for sequential prefetch and other I/O calculations. Use the most prominent 16 KB buffer pool here, even if several are used.

The BUFF16K default value is 00000250 buffers.

**BUFF32K nnnnnnnn**
Set the BUFF32K system parameter to the total number of 32 KB buffers in the 32 KB buffer pool of the dominant application. This value is used to set the number of pages per block for sequential prefetch and other I/O calculations. Use the most prominent 32 KB buffer pool here, even if several are used.

The default value is 00000100 buffers.

**BUFFERS nnnnnnnn**
Set the BUFFERS system parameter to the total number of 4 KB buffers in the 4 KB buffer pool of the dominant application. This value is used to set the number of pages per block for sequential prefetch and other I/O calculations. Specify the most prominent 4 KB buffer pool here, even if several are used.

The default value is 00002000 buffers.
COMMENT  remarks
You can insert a comment anywhere in a parameter file. All remarks are
transferred to the output report. To use this parameter, type COMMENT
followed by your remarks.

COSTING  nnnnnnnn
The COSTING system parameter indicates the maximum monetary value, in
your currency, that is permitted before the SQL statement is flagged as
exceeding the monetary limit. The maximum value is 999999999. A setting of 0
results in no monetary limit.

The default value is 15.

COSTQUN  nnnnnnnn
The COSTQUN system parameter indicates the maximum number of QUNITs
that are permitted before the SQL statement is flagged as exceeding the QUNIT
limit. The maximum value is 999999999. A setting of 0 results in no QUNIT
limit.

The default is 200.

Reminder: CPUTIME, ELAPSED, IOCALLS, COSTING, and COSTQUN are
limits for these resources. SQL PA flags all activity that exceeds the
limits set by these parameters. If no parameters are specified, SQL
PA assumes that there is no limit for that category.

CPUCOST  nnnn.nnn
Set the CPUCOST system parameter to the cost of one hour of CPU time, in
the national currency, for this configuration. This parameter is used by SQL PA
for charge back and financial cost determination. SQL PA

The default value is 500 per hour for processor time.

CPUTIME  nnnnn
The CPUTIME system parameter indicates the maximum CPU time, in
seconds, that is permitted before the SQL statement is flagged as exceeding the
time limit. The maximum value is 86400, which is equivalent to 24 hours. A
setting of 0 results in unlimited CPU time.

The default value is 10.

CURRSYM  a
Set the CURRSYM system parameter to the single character symbol that
represents the national currency that corresponds to the MONEYIS parameter.
CPUCOST, IOSCOST, and TIMCOST are used to calculate the total monetary
value of a transaction, based on your charge back costing guidelines. Setting
one or more of these parameters to zero, or leaving out, changes the cost
algorithm accordingly. For example, if your installation does not charge for
connect time, the costing would reflect only processor and I/O charges.

The default value is the dollar sign, $.

DATASHR  nnn
The DATASHR system parameter specifies the average percentage of the DB2
workload that participates in data sharing, from 0 to 100 percent. This
parameter is used with the DSGROUP parameter to adjust the overall
processing power of the system to reflect the excess processing time caused by
data sharing. The result of using these parameters is a more accurate cost
assessment.

The default is 0, no data sharing.
**DSGROUP**

The DSGROUP system parameter indicates the number of members in the data sharing group. This parameter is used with the DATASHR parameter to adjust the overall processing power of the system to reflect excess processing time caused by data sharing. The result of using these two parameters is a more accurate cost assessment.

The default is 0, no data sharing.

**DYNAMIC YES | NO**

Set the DYNAMIC system parameter to specify whether dynamic statement caching is used. Caching dynamic SQL statements can reduce the excess processing time associated with preparing dynamic SQL for subsequent operations. The values are YES (the system uses Dynamic Statement Caching) or NO.

The default value is NO.

**ELAPSED**

The ELAPSED system parameter indicates the maximum elapsed time, in seconds, that is permitted before the SQL statement is flagged as exceeding the elapsed time limit. The maximum value is 86400 which is equivalent to 24 hours. A setting of 0 results in unlimited elapsed time. For example, a value of 300 would be equivalent to 5 minutes (5 minutes * 60 seconds/minute = 300 seconds).

The default value is 60.

**ENGINES**

The total number of engines in this processor complex. For example, the IBM 9672-Z87 has eight engines.

The default value for a single engine processor is 00001.

**ESACOMP YES | NO**

The ESACOMP system parameter specifies whether this computer has a hardware data compression feature. For table spaces that are data compression candidates, the choice of hardware or software significantly affects the path length of encoding and decoding rows.

The default value is NO, implying software compression.

**ESASORT YES | NO**

The optional ESASORT parameter specifies whether the DB2 hardware assisted sort facility or its equivalent, is available on this processor. The DB2 sort algorithms for SQL PA are influenced by this parameter.

The default value is NO.

**FASTTBL YES | NO**

Use this parameter to indicate whether to bypass the explain plan processing. When FASTTBL is NO and the TABLES function is chosen, EXPLAIN is run on the statements. When FASTTBL is YES, and the TABLES function is chosen, processing bypasses the EXPLAIN, and uses the catalog queries to gather the needed information.

**IOCALLS**

The IOCALLS system parameter indicates the maximum physical I/O calls that are permitted before the SQL statement is flagged as exceeding the physical I/O limit. The maximum value is 999999999. A setting of 0 results in unlimited physical I/O.

The internal default value is 1000. The maximum value is 999999999.
**IOSCOST**  
*nnnn.nnn*  
Set the IOSCOST system parameter to the cost of 1000 I/O calls, in the national currency, for this configuration. This parameter is used by SQL PA for charge back and financial cost determination.

The default value is 10.00 per 1000 I/O.

**LPARENG**  
*nnnn.nnn*  
The LPARENG system parameter specifies the number of engines from the processor that is dedicated to the logical partition on which DB2 is running. You can specify the value in terms of partial processors, such as 1.85. This parameter specifically describes when a single physical processor complex is logically divided into multiple system images. Use this parameter to reflect the logical system image on which the DB2 target host system is running, and use it to isolate the maximum single engine speed.

The default value is 0001.000 (one engine).

**MONEYIS**  
*aaaaaaaaa*  
Set the MONEYIS system parameter to the national currency, in descriptive form, such as DOLLARS, POUNDS, STERLING, DRACHMA, or KRONA.

The default value is DOLLARS.

**SETPLAN**  
YES | NO  
The SETPLAN system parameter authorizes the use of the USEPLAN user parameter to indicate the qualifier of the plan table if its value is set to YES. If this parameter is set to NO, the value of USEPLAN is ignored, and the owner of the plan table defaults to the normal qualification. Using this parameter, you can control whether USEPLAN is operational under the batch, TSO, QMF, or stored procedure environments.

The default value for SETPLAN is NO. The default value for USEPLAN is +OFF+. That value (or blanks) also negates the effects of USEPLAN, regardless of the SETPLAN setting. SETPLAN is an optional solution intended for sites which restrict PLAN_TABLE access. When possible, use the generic ANLUSER1-n tables for processing.

The parameter is set in the data set identified by the ANLCNTL DD statement in batch and by the system parameters data set identified on the Parameter Data Sets panel for TSO. The default USEPLAN value is +OFF+, and it is not activated unless the SETPLAN YES parameter is also included in the user parameters data set member. If you implement this option for all TSO users, then the SANLPARM members, which constitute the DB2 Target Host selections, must be modified to include SETPLAN YES.

**Note:**

1. SETPLAN is an optional solution for those sites that need it. When possible, use the generic ANLUSER1-n tables for their processing. To comply with your security guidelines for SQL PA secondary authorization IDs, you might want to create an ALIAS for the generic ANLUSER1-n tables.

2. The default value for USEPLAN is +OFF+. That value (or blanks) also negates USEPLAN's effects, regardless of the SETPLAN value.

The default value is NOT USED.

**SUBSyst**  
*aaaaaaaaa*  
This parameter specifies the DB2 subsystem name, up to eight characters long, on this machine. Normally, DB2 is called DSN on most processors, but it is
possible to run several copies of DB2 on the same machine. If you run more than one copy of DB2, you must define a separate configuration file for each DB2 subsystem. It is not necessary to specify SUBSYST if your DB2 subsystem’s name is DSN.

The default value is DSN.

**SUBVERS aaaaaaaaa**

This parameter represents the DB2 version that you are using at your installation. Acceptable values are V8COM, V8NFM, V9COM, and V9NFM. V#COM and V#NFM differentiate between conversion mode and new function mode. This parameter is primarily used in the TSO Target Hosts library’s members, hiqual.SANLPARM, to allow the ANLALL CLIST a way to select alternate DB2 subsystems or program modules for a single TSO installation that accesses multiple DB2 subsystems at different DB2 levels.

It is also used to select the appropriate catalog access host variables for optimum performance and must always be specified.

**TIMCOST nnnn.nnn**

Set the TIMCOST system parameter to the cost of one hour of connect time, in the national currency, for this configuration. This parameter is used by SQL PA for charge back and financial cost determination.

The default value is 10 per connect hour.

**Note:** When PRECISE YES is used, only the processor time provided by the optimizer is used for cost assessment. I/O and elapsed time are free in this case.

**USRPARM YES | NO**

The USRPARM system parameter authorizes the use of the user-level user parameters.

The default value is NO.

---

**ANLPARM user parameters**

The following information describes the ANLPARM user parameters.

When running in batch mode, these parameters are stored in the file specified in the ANLPARM DD statement. They are expected to be modified by most SQL PA users, customized to each particular run. When using TSO, these parameters can be found in the SQL PA user parameter file of the Parameter Data Sets panel. Most of these parameters are used during the online execution, but a few of them are overridden by the values specified on the panels during invocation, such as DBRMKEY, DEGREES, QUALIFY, and USEPLAN.

**ADVISOR YES | NO | ALL**

The ADVISOR user parameter displays additional information about the performance and design of each query and about the tables and indexes that are accessed by the query. Warnings and Alerts are always displayed, even when ADVISOR is set to NO. Notes and Recommendations are displayed when ADVISOR is set to YES, which can reveal potential performance tuning opportunities. A value of ALL is ideal for those who want to receive guideline messages and confirmation that tasks have concluded successfully.

The SQL Advisor writes to the Explain and Detail Trace files, and is integrated with the TSO and batch reporting structure.

The default value is NO.
**ALLPART YES | NO**

Set the ALLPART parameter to YES to be notified when a partitioned table space scan reads all partitions in the table (the limited range scan is not used). Typically, reading all partitions in a table is not efficient.

The default value is YES.

**BUFFHIT nnn**

The BUFFHIT parameter indicates the ratio of physical to logical I/O, expressed as a percentage from 0 to 100. Specify a value for the percentage of pages that are found in the buffer pool (that is, for which disk I/O is not necessary).

The default value is 000, to indicate that 100% of I/O goes to DASD.

**COMMENT remarks**

You can insert a comment anywhere in either parameter file. All remarks are transferred to the output report. To use this parameter, type COMMENT followed by your remarks.

**CONNECT aaaaa**

The CONNECT user parameter specifies the expected application connection to DB2 when the SQL is started in production. This value helps SQL PA account for excess processing that is caused by various attach facilities, including thread management and application processing. The acceptable values are:

- IFP (IMS Fast Path)
- WFI (IMS Wait For Input)
- MPR (IMS Message Processing Region)
- BMP (IMS Batch Message Program)
- CICS (Online CICS/VS)
- SPUFI (Online TSO SPUFI)
- QMF (Online TSO QMF)
- DL1 (DL/1 Batch Program)
- CAF (Batch Call Attach Facility)
- DSN (Batch TMP with DSN CLIST)
- RRSAF (Recoverable Resource Manager Services Attach Facility)
- DRDA (remote connections)
- NONE (do not include any attach processor usage in cost)

The default value is CAF.

**DBRMKEY search-pattern**

The DBRMKEY parameter is only applicable when using the batch interface. To process multiple DBRMs using the ISPF interface, specify the search-pattern in the DBRM member name field on the Process DBRMs panel.

The DBRMKEY parameter provides a search pattern for selecting multiple DBRMs from the DBRM library pointed to by the ANLIN DD statement in a batch run. You can process up to 200 DBRMs in a single run.

The search pattern can contain up to 8 characters, including the wildcard characters, a percent sign (%) and asterisk (*), as shown in the following examples:

```
bbbbbbbb
```

A value of all blanks causes all members in the library to be selected.
An asterisk (*) in the first position also causes all DBRMs in the library to be selected.

**ABC**  Causes all members whose names start with ABC in the first three characters to be selected. Other character positions in the names are not relevant.

**member**  Causes the DBRM member whose name is **member** to be selected from the library.

%%%ABC%%%

Causes members whose names have any character in the % positions and ABC in the third, fourth, and fifth positions to be selected. The length of a member name **must match exactly** the length of the pattern. For example, with DBRMKEY %%%4%%%, the DBRM named ABC4SSS would be selected, but a DBRM named ABC4QQ would not be selected.

**+OFF+**  This special DBRMKEY value turns off the processing of multiple DBRMs, while allowing the DBRMKEY parameter to remain in the user’s parameter list.

There are three requirements to process multiple DBRMs in a batch run of SQL PA:

1. The DBRMKEY parameter must be present with a value other than **+OFF+**.
2. An ANLWORK DD statement must be in the batch JCL with DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000) to provide temporary workspace.
3. The ANLIN DD statement must allocate the required DBRM library, such as DSN=MY.DBRMLIB,DISP=SHR.

**Note:** When processing multiple DBRMs, do not specify a member name in the ANLIN DD statement unless the DBRMKEY value is **+OFF+**. Otherwise the results are unpredictable.

**DEGREES ANY | ONE | 1**

Set the DEGREES user parameter to specify whether parallel processing is considered during the statement evaluation when the batch interface is used. Set DEGREES to **ANY** when parallel processing is enabled. Set DEGREES to **ONE** when parallel processing is not enabled.

**ANY**  If set to **ANY**, SQL PA informs the optimizer that parallel processing is an option.

**ONE | 1**  If set to **ONE** or 1, SQL PA tells DB2 to ignore any parallel cases, even if the query could benefit from parallel processing (a table space scan against a partitioned table, for example).

The default value is 1.

**DELIMIT QUOTE**

COBOL programs have mutually exclusive precompile options: QUOTESQL or APOSTSQL, which determine whether the SQL uses double quotation marks ("\") around literal strings instead of single quotation marks or apostrophes ('). Only the COBOL language permits this choice, and most DB2 sites standardize using the single quotation mark during installation. However, if you compile
COBOL programs with the double quotation mark delimiter in this environment, DB2 Prepare/Explain rejects the SQL with -206 SQL Code, making it unrecognizable, and thus SQL PA would be unable to provide a cost. If you choose to use double quotation marks to delimit SQL, and the -206 SQL Code is returned by DB2, you can correct the situation by using the DELIMIT QUOTE parameter. This user parameter instructs SQL PA to transform and temporarily replace double quotation marks with single quotation marks for the Prepare/Explain process, eliminating the -206 error from DB2. This parameter does not change or affect your SQL in any way. Only the processing under c is modified. SQL with single quotation marks are unaffected by this parameter, can be intermixed, and process normally.

**DSN8EXP YES | NO**
Set the DSN8EXP user parameter to specify whether EXPLAINs are run by using the DB2-supplied stored procedure (DSNAEXP) or by embedded SQL. If the value is set to YES, DSNAEXP is used if it is installed and available; otherwise, DSN8EXP is used.

The default setting is NO.

**DSPPARM YES | NO**
The DSPPARM user parameter determines whether the SQL PA parameters and their values are displayed at the top of the enhanced explain and detail trace reports.

The default setting is YES.

**DSPVARS YES | NO**
The DSPVARS user parameter determines whether the host variable name s or parameter markers are displayed in SQL statements in the cost reports.

Restriction: DSPVARS is limited to SQL statements with a length less than 32720.

**DSPWCOC YES | NO**
For DBRMs, specifies whether WHERE CURRENT OF statements are shown in the Query Limits report. The WHERE CURRENT OF clause can falsely cause QUNIT values that exceed limits set for acceptable query statements. Set this parameter to NO, to have WHERE CURRENT OF statements excluded from the Query Limits report.

**EEECALL SPA | EEEPATH**
This parameter notifies SQL PA of the source of the input SQL statements. If you set this parameter to SPA, SQL PA processes data from the ANLIN files or a DBRM library. When ANLIEE calls SQL PA to do its analysis, the data source is noted, so SQL PA knows the source of the input statements.

SPA Normal DB2 SQL Performance Analyzer operation, or retro explain (EXPLAIN OLD).

EEEPATH
The SQL comes from the catalog (SYSSTMT, SYSPACKSTMT) and the OLD access path data is in the EEEPATH table, along with the corresponding SQL in the userid.ANLEEE.SQL file.

The default value is SPA.

**EXPLAIN NEW | OLD**
The EXPLAIN parameter indicates how explain information is gathered when
running ANLSQLPA in batch. Use the EXPLAIN parameter to request that a new EXPLAIN be run to generate the information, or that existing information be read from the EXPLAIN tables.

**NEW** Creates new entries in the EXPLAIN tables.

**OLD** Uses existing EXPLAIN tables. Set the USEPLAN parameter to identify the owner of the EXPLAIN tables and set QUERYNO and STOPQNO to limit the statements read from those tables.

The default value is NEW.

**HPOOLRD** `nnn`  
Beginning with Version 8, DB2 does not support Hiperpools – if specified, it is ignored.

**INLISTS** `nnn`  
The INLISTS user parameter specifies the limit of the number of elements that are shown in an IN (LIST) predicate. If the maximum number of elements is exceeded, DB2 SQL Performance Analyzer issues a notification message.

The default value is 10 elements.

**ISCANPG** `nnnnnnnn`  
The ISCANPG user parameter indicates the limit of the number of index leaf pages that are read with a matching index scan. If the maximum number of pages is exceeded, SQL PA issues a notification message.

The default value is 1,000 pages.

**IXUPDAT** `nnn`  
The IXUPDAT user parameter indicates the limit of the number of indexes that can exist on a table that is being updated. If the maximum number of indexes is exceeded, SQL PA issues a notification message.

The default value is 5 indexes.

**JOINTAB** `nnn`  
The JOINTAB user parameter indicates the limit of the number of tables that can be joined. If the maximum number of tables is exceeded, SQL PA issues a notification message.

The default value is 10 tables.

**KEEPPLAN** **YES** | **NO**  
The KEEPPLAN user parameter indicates whether plan records that are owned by the value of the USEPLAN parameter are retained when SQL PA runs in batch. SQL PA deletes all plan records at the conclusion of each run, and it deletes all plan records with QUERYNO greater than 100 M on startup. If this parameter is set to YES, the deletion of plan records in your permanent plan tables is prevented. This parameter does not affect online operation, and it does not affect the generic plan tables.

You are able to access this data until the next SQL PA startup.

**Restriction:** This parameter operates only in batch mode and applies only to USEPLAN directed or explain tables owned by the current authorization ID. It does not operate for generic PLAN_TABLEs or work under TSO ISPF.

**MATCOLS** `n.nn`  
The MATCOLS user parameter limits the number of columns in an index that are used in a matching index scan. The parameter value expresses the fraction
of columns in the index key. For example, the default value of 0.50 indicated
that at least half of the columns in the index key must be used in a matching
index scan. If not, SQL PA issues a notification message that indicates that the
index is not being used to its fullest potential.

The default value is 0.50 (match half the columns or more).

**MESSAGE YES | NO**

The MESSAGE user parameter determines whether the message text will be
included in the SQL PA output. If this parameter is set to NO, only the
message ID will be displayed.

The default value is YES which includes all of the message text in the output.

**NEWROTA n.nnn**

The NEWROTA user parameter specifies the average rotational delay (half of a
revolution, also called latency) in seconds for the definition of a new storage
device. This parameter is effective only when the NEWSTOR parameter is also
set or when the STORAGE user parameter is set to NEWDSK.

**NEWSEEK n.nnn**

The NEWSEEK user parameter specifies the average seek time in seconds for
the definition of a new storage device. This parameter is effective only when
the NEWSTOR parameter is also set or when the STORAGE user parameter is
set to NEWDSK.

**NEWSTOR newname**

The NEWSTOR user parameter indicates that you are defining a new storage
device. This parameter must be followed with the NEWSEEK, NEWROTO, or
NEWXFER parameters to define the new device. Either this parameter or a
STORAGE NEWDSK setting indicates that new disk storage is being defined.

**NEWXFER nnnnn.n**

The NEWXFER user parameter specifies the average transfer rate of the device
over the channel in kilobytes per second. This parameter is effective only when
the NEWSTOR parameter is also set or when the STORAGE parameter is set to
NEWDSK.

The default value is WDSK.

**NLSCODE +OFF+ | KOR | JPN | CHS | CHT | ON**

The NLSCODE parameter enables processing SQL that includes DBCS or
Mixed Data names, such as that found in the following national languages:

- KOR (Korean)
- JPN (Japanese)
- CHS (Chinese)

If you specify +OFF+, SQL is converted to uppercase before processing, a
translation that might alter the DBCS and mixed data names. When one of
these national languages is selected, this translation to uppercase does not take
place. The parser scans for and identifies shift-out and shift-in characters,
which bracket the DBCS and mixed data names and leaves all characters
within them unchanged while processing the SQL. This process allows DB2 to
find the referenced DBCS and mixed data columns and tables. A value of ON
turns off lowercase to uppercase translation.

The default value is +OFF+.

**NONIXPG nnnnnnn**

The NONIXPG user parameter indicates the limit of the number of index leaf
pages that are read with a non-matching index scan. If you exceed the number of pages specified, SQL PA issues a notification message.

The default value is 5,000 pages.

**NONONIX YES | NO**

Set the NONONIX parameter to YES to be notified of every non-matching index scan operation.

The default value is NO.

**NOSTATS YES | NO**

Set the NOSTATS user parameter to YES to be notified when the default was used for the statistics of any table, index, table space, or column because the catalog statistics were not set.

The default value is YES.

**NOTSCAN YES | NO**

Set the NOTSCAN user parameter to YES to be notified of every table space scan operation.

The default value is NO.

**NUMBERS YES | NO**

The NUMBERS user parameter specifies whether the input file was created with NUMBERS ON.

The default value is YES.

**OBJECTS YES | NO**

The OBJECTS user parameter determines whether the list of objects (tables and indexes) that are used will be included in the SQL report. If this parameter is set to YES, the list of objects is displayed. If this parameter is set to NO, the list of objects is not displayed.

This list is appended to the userid.ANLOUT.SQL file, which also contains a list of the SQL statements that were processed in the run. The ANLOUT DD card allocates this file in batch. To display the contents under TSO, select option 5 on the Reports menu (Figure 41 on page 133).

The default value is NO.

**PRECISE YES | NO | ALL**

The PRECISE user parameter provides an extra level of precision to the cost estimate. If this parameter is set to YES, DB2 SQL Performance Analyzer replaces its internal cost estimate and QUNIT result with the values from DB2. It also uses the path length of the optimizer in calculations. If this parameter is set to ALL, the cost estimate includes additional factors and the Predicate Analysis report is generated.

**YES**

When PRECISE is set to YES, the DB2 SQL Performance Analyzer cost analysis attempts to replace its internal cost estimate with the DB2 optimizer's own processor time estimate, drawn from the optional Explain table DSN_STATEMENT_TABLE and the value for PROCMS, or processor milliseconds. Also included in this table is the PROCSU or Service Units estimate, which replaces the value DB2 SQL Performance Analyzer calculates for QUNITS.

PRECISE YES causes the optimizer's estimate of path length to be used in calculations. However, be aware that this path length is less extensive than SQL PA's because it only considers the data access portion of the processing. Especially significant are Category B estimates, where the cost
estimate is definitely incomplete due to referential integrity, triggers, user-defined functions, and so forth. For more information about the Category B classification, see *DB2 UDB for z/OS V8 Application Programming and SQL Guide*, SC18-7415.

The values returned from PRECISE YES can also be used to govern dynamic queries through the IBM Resource Limit Facility, or RLF. This is accomplished by using the PROCSU (QUNITS) values output from DB2 SQL Performance Analyzer evaluations to populate the RLFASUWARN and RLFASUERR columns in the DSNRLST table. Then, a threshold for warnings and errors are set in Service Units for the actual running of the dynamic queries evaluated by DB2 SQL Performance Analyzer.

ALL

PRECISE ALL produces DB2 SQL Performance Analyzer cost estimates that include additional factors, extending the cost analysis. With PRECISE ALL, the Predicate Analysis report is generated. This analysis becomes a part of the Enhance Explain and Detailed Trace reports, providing filter factors, predicate processing attributes, and other key insights.

The default value for PRECISE is YES.

PREDPCT n.nn

The PREDPCT user parameter limits the fraction of the number of total rows to be retained after predicate filtering. When this limit is exceeded, a notification message is issued that indicates that the predicates are not restrictive. For example the default value of 0.15 states that if more than 15% of the rows pass filtering, DB2 SQL Performance Analyzer issues a message.

The default value is 0.15, or 15% of all rows.

PROCESS +OFF+ | ASIS | NOSEQ

Set this parameter to ASIS to turn off the automatic DB2 version recognition capability. If you have coded a VERSION parameter that specifies a DB2 release that is different from the actual version under which the program is running, the VERSION parameter is reset automatically unless the ASIS value is set. When the PROCESS parameter is set to +OFF+, ANL2021W is displayed when the VERSION parameter is reset. If you want DB2 SQL Performance Analyzer to process the Version parameter ASIS, regardless of the actual version (computing certain costs based on a particular DB2 release level) then the PROCESS ASIS parameter can be coded, which turns off auto-version recognition.

Some DBRM modules are created or modified by vendor software products, and sometimes these modifications can combine several packages or plans into one module, or create DBRMs with statement sequence numbers out of sequential order. DB2 SQL Performance Analyzer expects the statement numbers in DBRMs to always be sequential (as the prepared and explained plans are ordered by QUERYNO) and generates ANL2003E Synchronization Errors when this is not the case. To eliminate these errors and process the out-of-sequence DBRM successfully, specify PROCESS NOSEQ to indicate that the DBRM contains statement numbers that are not sequential. When coded, PROCESS NOSEQ causes DB2 SQL Performance Analyzer to assign a unique query numbering scheme that avoids statement sequence errors and handles these DBRMs correctly.

The PROCESS ASIS and PROCESS NOSEQ parameters are independent and can both be present in the ANLPARM file. They are listed as separate options on the TSO parameters panel as well.
QLIMSQL YES | NO
Set the QLIMSQL user parameter to YES to display an SQL statement that received a negative SQLCODE in the QLIMITS report.

The default value NO.

QUALIFY qqqqqqqq
The QUALIFY user parameter specifies the qualifier that will be used to qualify unqualified objects in an SQL statement. If no value is specified, the current SQL ID will be used. To qualify unqualified tables when using the ISPF interface, see "Qualifying unqualified objects in plans and packages" on page 149 and "Qualifying unqualified objects in DBRMs and flat files" on page 150.

This optional parameter supplies the default high-level name qualifier for all single-level table and view names when using the batch interface to explain DBRMs and flat files. It is used as the CREATOR prefix for single-level names. For example, if the SQL contained the following statement:

```
SELECT * FROM L1000 WHERE CIKEY > 25
```

SQL PA must know the high-level qualifier for the L1000 table, currently unspecified in the SQL. If QUALIFY is set to 'TDT690', SQL PA interprets the query as follows:

```
SELECT * FROM TDT690.L1000 WHERE CIKEY > 25
```

The parser section of component of DB2 SQL Performance Analyzer automatically adjusts. If no value is specified for QUALIFY, it defaults to the current user's SQLID.

Restriction
QUALIFY is limited to SQL statements with a length less than 32720.

Long Qualifiers in Version 8
High-level qualifiers are limited to 128 characters in DB2 Version 8 NFM. To specify a long name as a qualifier, continue the name beyond the normal eight characters allotted for the QUALIFY parameter and continue with another QUALIFY parameter as the next record to finish up the name if necessary. You can also use a + on the second card as a continuation alternative anywhere in columns 1-7, and column 8 is always blank. DB2 SQL Performance Analyzer interprets the first blank encountered as the end of the qualifier name.

Specifying alternative EXPLAIN tables
Some installations do not (or cannot) modify the DSN3@ATH authorization exit to add the generic ANLUSER1-n authorization IDs, which own the reusable PLAN_TABLEs and DSN_STATEMNT_TABLEs. There are several options available to solve this authorization scheme, including using a PLAN_TABLE under the primary authorization ID, or explicitly granting access to the ANLUSERx tables, or using other generic tables under a secondary authorization ID.

You can also use the USEPLAN authid parameter, which specifies the particular high-level qualifier of the Plan and DSN_STATEMNT_TABLEs to be used by this run. To activate the parameter, you must also set the SETPLAN parameter to YES, which authorizes users to select their own high-level qualifier. If SETPLAN is NO or is not present, DB2 SQL Performance Analyzer ignores the USEPLAN, and nothing changes.

QUERYNO ALL | nnnnnnnnn
Use the QUERYNO parameter to specify the plans to be analyzed in the table.
ALL

Set the QUERYNO parameter to ALL to analyze all of the plans in the table.

nnnnnnnnn

To specify a range of plans in the table to analyze, set the QUERYNO parameter to the beginning statement number and set the STOPQNO parameter to the end statement number.

Both of these parameters are optional. If you do not specify any values, every plan in the PLAN_TABLE source is analyzed. The QUERYNO format is nine characters. DB2 SQL Performance Analyzer begins at plan 100,000,001 for its insertions to the PLAN_TABLE for NEW plan records.

The default value is ALL.

REFRESH ANY | ALL | NO

The REFRESH user parameter allows the optimizer to consider materialized query tables (MQTs) as candidates for access path selection if the value is set to ALL or ANY. If this parameter is set to NO, MQTs are not considered.

The default value is NO, meaning that MQTs cannot be considered by the optimizer during access path selection.

REPORTS LOG | REP | DET

The REPORTS user parameter specifies the level of reporting. Specify one of the following values:

LOG

Turns on cost reporting, providing a summary of query costs, plus any warning messages. This report is always available.

REP

Turns on the enhanced explain reporting.

DET

Provides detailed trace information. Specifying this parameter triggers the Detail Trace Report that contains information about each step of operation, with the SQL PA estimates of DB2 resource consumption at the most granular level.

LOG is the default value when you leave this parameter blank.

The QMF Interface Program, ANLGOV1, uses two special additional values for this parameter, MAX or ANY. Also, the stored procedures, ANLPRCR and ANLPRER use a value of STP in this parameter. The SEEMSGS parameter applies only to QMF Interface Program, ANLGOV1, and is invalid in the ANLPARM file.

The default value of the QMF Interface Program is MAX, which is also the minimum level of reporting that you can select under TSO.

RETCODE YES | NO

The RETCODE user parameter governs whether the return code reflects the warnings, error messages, or cost overruns from SQL PA.

NO

A value of NO leaves program return codes unchanged, so they continue to reflect normal processing of SQL PA programs.

YES

A value of YES causes the program return codes to reflect the highest message level achieved within each program. When RETCODE is set to
YES, values of 4, 8, 12, or 16 are observed as program return codes, indicating that warnings, error messages or cost overruns have been generated by the SQL that was examined. This is useful in identifying that further study is warranted when processing jobs with multiple DBRMs, or SQL migrating from Test to Production. Regardless of the RETCODE value set, the highest message level achieved is always displayed in the SYSOUT (or userid.ANLCOST.LOG in TSO) for DB2 SQL Performance Analyzer’s ANLSQLPA program.

The default value is NO.

**RETPLAN YES | NO**

Only the explain-capable ANLPRER stored procedure recognizes the RETPLAN parameter. RETPLAN is passed to this procedure by the user parameter list that is embedded in the ANLSTER sample program. In addition to the usual cost estimates and warning flags, when RETPLAN is set to YES, it signals the stored procedure to return Explain plan data and catalog statistics to the calling program. You can use the sample programs as a guide when writing your own stored procedure that calls programs.

The default value is NO.

**SHOWALT YES | NO**

Set the SHOWALT user parameter to YES to display information about alternate indexes that are not selected by the optimizer in the Enhanced Explain and detail trace reports. The information is presented for access paths of I, N, and R only, and is not presented for multiple index or direct access using row ID. The information includes pertinent catalog statistics about the size and shape of each alternate index, and information about the index key’s columns.

The default value is NO.

**STOPQNO**********

You must use this parameter with the QUERYNO parameter. The STOPQNO parameter must be specified after the QUERYNO parameter.

**********

To specify a range of plans in the table to analyze, set the QUERYNO parameter to the beginning statement number and set the STOPQNO parameter to the end statement number.

Both of these parameters are optional. However, if you do not specify any values, every plan in the PLAN_TABLE source is analyzed.

**STORAGE ddddd**

The STORAGE user parameter reflects the most prominent DASD storage medium for an application’s databases. If your system uses several device types, select the device that holds most of your databases. This value is used for I/O and elapsed time estimates. The acceptable values include 3390, 3390-1, 3390-2, 3390-3, 3390-9, 6390, 6390-3, 6390-5, 6390-9, 7390-1, 7390-2, 7390-3, or 7390-F disk device types, and solid-state devices like the 4305, 4380, 6110, 6110-1, 6110-2, 6110-4, 6680-1, 7900, ICEBRG, STK, EMCX, EMCBAR, EMCELST, RAMAC, RAMAC2, and RAMAC3, E SSE20, ESSF20, ESS800, and ESS80T (turbo). If several device types are on your system, select the device which holds most of your databases.

The STORAGE default value is the 3390-3 disk device.
SYNONYM
The value of the SYNONYM parameter indicates the creator of synonyms for
the SQL statement that is being explained when SQL PA is being run in batch.

TSCANPG nnnnnnnnnn
The TSCANPG user parameter indicates the limit of the number of data pages
that are read with a table space scan. If the maximum number of pages is
exceeded, DB2 SQL Performance Analyzer issues a notification message.

The default value is 50,000 pages.

TURNOFF nnnn BYPASSRC
The TURNOFF user parameter excludes specified messages from the SQL PA
output. You must enter each TURNOFF parameter on a separate line. You can
specify up to 100 TURNOFF parameters.

Specify the 4-digit numeric message number to turn off the message. For
example, TURNOFF 6011 would exclude message ANL6011I from the output.

To ensure that the return code is not set for the same message number that you
specify in the TURNOFF parameter, use the following option:

BYPASSRC
This keyword is optional. When you specify the BYPASSRC keyword,
the return code is not set for the same specified message number. For
example, TURNOFF 6011 BYPASSRC would exclude message ANL6011I
from the output, and the return code would not be set.

The BYPASSRC keyword is used for batch processing only.

The TURNOFF parameter and the BYPASSRC keyword are available only for
ANLxnnn messages, where x is 5, 6, or 7, and nnn is the last three digits of the
message number. However, the TURNOFF parameter is not available for the

To avoid these message, increase the appropriate limits.

The default is to show all messages.

USEPLAN authid+OFF+
This optional parameter specifies a specific PLAN (and DSN_STATEMNT)
table under this authorization ID. Omit or set the parameter to +OFF+ when
this option is not used. The parameter is activated only when the
corresponding SETPLAN parameter is set to YES in the ANLCNTL parameters
for this run, and it is only used with the batch interface.

When using the ISPF interface, the PLAN (and DSN_STATEMNT) table used
corresponds to the current SQL ID as indicated on the panels. This option is
also on the TSO panels when the current SQL ID is set to +OFF+, and can be
available under QMF (using the running SQLID). If active under QMF, the
SQLID is changed to the authorization ID yo want by issuing the SET
CURRENT SQLID authorization ID statement under QMF.

The default value is +OFF+.

VERSION aaaa
This parameter indicates the version and release of DB2 that you are using so
that SQL PA can select the appropriate path length values. Specify V9R1,
V10R1, or V11R1. The default value is V10R1 for DB2 10.

VIADRDA location-name
This optional parameter routes processing to another DB2 system. The
location-name (a 16-character field) is any valid location specified in the local
SYSIBM.LOCATIONS table. The parameter can be left in ANLPARM with a
value of +OFF+ to make it non-operational. Because of the size of the parameter field (16 characters), the comments for this parameter must start in column 31:

**VIADRDA DALLAS-CQ ROUTE QUERIES TO DALCQ SYSTEM**

Because of the size of the parameter field (16 characters) the comments for this parameter must start in column 31. For example:

```
REPORTS ALL  SELECT REPORT LEVEL: YES, EXP OR ALL
VIADRDA DALLAS-CQ  ROUTE QUERIES TO DALCQ SYSTEM
```

The SQL PA user parameters are specified in the file allocated to ANLPARM ddname in batch mode, or taken from the user parameters file on the SQL PA Parameter Data Sets panel. The following figure shows two examples of these parameters.

---

**#1.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value or Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORTS</td>
<td>REP THREE LEVELS OF REPORTING: LOGS, REPORTS, DETAILS</td>
</tr>
<tr>
<td>VERSION</td>
<td>V10R1 THE CURRENT DB2 RELEASE IS V10R1 (ANY MODE)</td>
</tr>
<tr>
<td>USEPLAN</td>
<td>TDT690 USE DIRECTED PLAN_TABLE UNDER TDT690 (OR '+OFF+')</td>
</tr>
<tr>
<td>STORAGE</td>
<td>ESS80T THE DASD FARM IS SHARK ESS MODEL 800 TURBO</td>
</tr>
<tr>
<td>BUFFHIT</td>
<td>015 15% OF PAGES ARE FOUND IN BUFFER POOL</td>
</tr>
<tr>
<td>CONNECT</td>
<td>SPUFI ATTACH OVERHEAD: CAF</td>
</tr>
<tr>
<td>DBRMKEY</td>
<td>ANL* PROCESS ANY MEMBERS STARTING WITH 'ANL' (OR '+OFF+')</td>
</tr>
<tr>
<td>QUALIFY</td>
<td>TDT690 DEFAULT HIGH LEVEL QUALIFIER =&gt; USE YOUR OWN AUTHID</td>
</tr>
<tr>
<td>PRECISE</td>
<td>YES USE OPTIMIZER CPU &amp; SERVICE UNIT COSTING YES</td>
</tr>
<tr>
<td>ADVISOR</td>
<td>ALL WARNINGS/ALERTS + RECS/NOTES=YES, GUIDES/GOOD=ALL</td>
</tr>
<tr>
<td>SHOWALT</td>
<td>YES SHOW ALTERNATE INDEXES NOT SELECTED BY DB2 OPTIMIZER</td>
</tr>
<tr>
<td>RETCODE</td>
<td>NO RETCODE USED FOR MESSAGE LEVEL REPORTING YES</td>
</tr>
</tbody>
</table>

**#2.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value or Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORTS</td>
<td>DET THREE LEVELS OF REPORTING: LOGS, REPORTS, DETAILS</td>
</tr>
<tr>
<td>SHOWALT</td>
<td>NO SHOW ALTERNATE INDEXES NOT SELECTED BY DB2 OPTIMIZER</td>
</tr>
<tr>
<td>VIADRDA</td>
<td>DALLAS-CQ ROUTE VIA DRDA CONNECT TO DALLAS SYSTEM CQ</td>
</tr>
<tr>
<td>VERSION</td>
<td>V7R1 THE CURRENT DB2 RELEASE IS V7R1</td>
</tr>
<tr>
<td>STORAGE</td>
<td>3390-3 THE DASD FARM IS 3390-3 DISKS</td>
</tr>
<tr>
<td>BUFFHIT</td>
<td>020 20% OF PAGES ARE FOUND IN BUFFER POOL</td>
</tr>
<tr>
<td>DEGREES</td>
<td>1 CONSIDER PARALLELISM FOR PARTITIONED TABLESPACES</td>
</tr>
<tr>
<td>CONNECT</td>
<td>DRDA ATTACH OVERHEAD: CAF</td>
</tr>
<tr>
<td>DBRMKEY</td>
<td>+OFF+ PROCESS ANY MEMBERS STARTING WITH 'ANL' (+OFF+)</td>
</tr>
<tr>
<td>PRECISE</td>
<td>YES EXTRA COST OPTIMIZATION FOR PRECISION YES</td>
</tr>
<tr>
<td>ADVISOR</td>
<td>YES WARNINGS/ALERTS + RECS/NOTES=YES, GUIDES/GOOD=ALL</td>
</tr>
<tr>
<td>RETCODE</td>
<td>YES RETCODE USED FOR MESSAGE LEVEL REPORTING YES</td>
</tr>
</tbody>
</table>

(Note: no QUALIFY parameter in sample #2, so it defaults to CURRENT SQLID)

---

**Figure 68. Examples of user parameters**

**ANLPARM user parameter settings for QMF**

QMF requires some specific settings in the ANLPARM user parameter list.

**SUBSYST**

SUBSYST must reflect the name of the DB2 subsystem (DSN is the default) for the CAF connections.

**REPORTS**

If REPORTS is set to MAX, SQL PA only shows costs for those queries that have exceeded one or more limits. MAX is the recommended setting.
If REPORTS is set to ANY, SQL PA shows the costs for any query processed. With the MAX option, SQL PA remains invisible unless the user exceeds a resource threshold.

**CONNECT**

CONNECT must be set to QMF.

**SEEMSGS**

If SEEMSGS is set to YES during initial testing or debugging, SQL PA shows informational and error messages encountered during processing. Many of these messages are also displayed by QMF.

If SEEMSGS is set to NO, these messages are not displayed by SQL PA. NO is the recommended setting.

---

**Easy Explain table contents**

Easy Explain depends on a number of DB2 tables.

**Topics:**

- "The EEEAUTH table"
- "The EEEWORK table"
- "The EEEDBRM table" on page 281
- "The EEEHINT table" on page 282
- "The EEEPLAN table" on page 282
- "The EEEPATH table" on page 283

**The EEEAUTH table**

The EEEAUTH table is used to specify the privileges for each Easy Explain user.

If you do not have an entry in the table, default privileges specified in the row with a blank authorization ID are used.

**Table 21. EEEAUTH table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authid</td>
<td>Char(8) Not Null</td>
<td>Authorization ID of the user</td>
</tr>
<tr>
<td>NLS</td>
<td>Char(1) Not Null</td>
<td>Indicator for National Language (Y/N)</td>
</tr>
<tr>
<td>Planinfo</td>
<td>Char(1) Not Null</td>
<td>Indicator for extended plan info. (Y/N)</td>
</tr>
<tr>
<td>FormatSQL</td>
<td>Char(1) Not Null</td>
<td>Indicator for formatting of SQL (Y/N)</td>
</tr>
<tr>
<td>Timestmp</td>
<td>Timestamp NNWD</td>
<td>Timestamp for creation or change for row</td>
</tr>
<tr>
<td>Remotexpl</td>
<td>Char(1) NNWD</td>
<td>Indicator for use of remote explain (Y/N)</td>
</tr>
</tbody>
</table>

**The EEEWORK table**

The EEEWORK table is used internally by Easy Explain to find all the indexes used by a given plan or package if a parameter of INDEX=Condensed has been specified.

**Table 22. EEEWORK table**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Char(16) NNWD</td>
<td>Location name for DB2 subsystem, or blank</td>
</tr>
</tbody>
</table>
A view has been defined on the EEEWORK table. This view called EEE_LOCATION is used for defining the database and table space names for the location of the PLAN_TABLE tables created by Easy Explain. Only the first three columns are used.

**Table 23. EEE_LOCATION view**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Char(16)</td>
<td>Location name for DB2 subsystem, or blank</td>
</tr>
<tr>
<td>DBname</td>
<td>Char(8)</td>
<td>Database name</td>
</tr>
<tr>
<td>TSname</td>
<td>Char(8)</td>
<td>Table space name</td>
</tr>
</tbody>
</table>

The EEEDBRM table

The EEEDBRM table is used internally by Easy Explain to store DBRM and package information. This information is used to build the Plan Information report.

**Table 24. EEEDBRM table**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>char (8)</td>
<td>DBRM name</td>
</tr>
<tr>
<td>plname</td>
<td>char (8)</td>
<td>Plan name</td>
</tr>
<tr>
<td>plcreator</td>
<td>char (8)</td>
<td>Plan creator name</td>
</tr>
<tr>
<td>precompdate</td>
<td>char (6)</td>
<td>Date for precompilation</td>
</tr>
<tr>
<td>precomptime</td>
<td>char (8)</td>
<td>Time for precompilation</td>
</tr>
<tr>
<td>sqlstmt</td>
<td>integer</td>
<td>Number of SQL statements in DBRM</td>
</tr>
<tr>
<td>hostlang</td>
<td>char (1)</td>
<td>Programming language</td>
</tr>
<tr>
<td>charset</td>
<td>char (1)</td>
<td>Character set (Alphanumeric or Katakana)</td>
</tr>
<tr>
<td>comma</td>
<td>char (1)</td>
<td>Comma or period</td>
</tr>
<tr>
<td>dec31</td>
<td>char (1)</td>
<td>Decimal 31 used at precompile</td>
</tr>
<tr>
<td>dbrmrel</td>
<td>char (1)</td>
<td>DB2 release at precompile</td>
</tr>
<tr>
<td>type</td>
<td>char (1)</td>
<td>DBRM or Package</td>
</tr>
<tr>
<td>storedby</td>
<td>char (8)</td>
<td>Authorization ID</td>
</tr>
<tr>
<td>timestamp</td>
<td>timestamp</td>
<td>Timestamp for storing row</td>
</tr>
<tr>
<td>pdsname</td>
<td>char(44)</td>
<td>Full MVS pds name for DBRM</td>
</tr>
<tr>
<td>collid</td>
<td>char(18)</td>
<td>Collection identification</td>
</tr>
<tr>
<td>version</td>
<td>varchar(64)</td>
<td>Version identification</td>
</tr>
</tbody>
</table>
The EEEHINT table

The EEEHINT table is used to save hint IDs for the miniplans generated by Easy Explain. The ID identifies each miniplan by a query number, a user ID, and a timestamp.

Table 25. EEEHINT table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Char(16)</td>
<td>Not Null Location of miniplan</td>
</tr>
<tr>
<td>Hint_ID</td>
<td>Integer</td>
<td>Not Null Unique number identifying miniplan</td>
</tr>
<tr>
<td>Creator</td>
<td>Char(8)</td>
<td>Not Null User ID for miniplan creator</td>
</tr>
<tr>
<td>Opthint</td>
<td>Char(8)</td>
<td>Not Null Specified optimization hint</td>
</tr>
<tr>
<td>Queryno</td>
<td>Integer</td>
<td>Not Null Query number taken from PLAN_TABLE table</td>
</tr>
<tr>
<td>Bind_time</td>
<td>Timestamp</td>
<td>NNWD Timestamp for creation of miniplan row</td>
</tr>
<tr>
<td>Progname</td>
<td>Char(8)</td>
<td>NNWD Name of program or package</td>
</tr>
</tbody>
</table>

The EEEPLAN table

The EEEPLAN table is used internally by Easy Explain to store plan information for plans explained by Easy Explain. The plan information is used to build the Plan Information report.

Table 26. EEEPLAN table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>char (8)</td>
<td>not null Plan name</td>
</tr>
<tr>
<td>creator</td>
<td>char (8)</td>
<td>not null Creator name</td>
</tr>
<tr>
<td>binddate</td>
<td>char (6)</td>
<td>not null Date for bind</td>
</tr>
<tr>
<td>bindtime</td>
<td>char (8)</td>
<td>not null Time for bind</td>
</tr>
<tr>
<td>boundby</td>
<td>char (8)</td>
<td>not null Authorization ID for binder</td>
</tr>
<tr>
<td>plsize</td>
<td>integer</td>
<td>not null Size of plan</td>
</tr>
<tr>
<td>avgsize</td>
<td>integer</td>
<td>not null Average size of plan sections</td>
</tr>
<tr>
<td>sqlstmt</td>
<td>integer</td>
<td>not null Number of SQL statements in plan</td>
</tr>
<tr>
<td>validate</td>
<td>char (1)</td>
<td>not null Validity checking</td>
</tr>
<tr>
<td>isolation</td>
<td>char (1)</td>
<td>not null Isolation level</td>
</tr>
<tr>
<td>valid</td>
<td>char (1)</td>
<td>not null Plan or package valid</td>
</tr>
<tr>
<td>operative</td>
<td>char (1)</td>
<td>not null Plan or package operative</td>
</tr>
<tr>
<td>acquire</td>
<td>char (1)</td>
<td>not null Acquire option</td>
</tr>
<tr>
<td>release</td>
<td>char (1)</td>
<td>not null Release option</td>
</tr>
<tr>
<td>qualifier</td>
<td>char (8)</td>
<td>not null Table qualifier</td>
</tr>
<tr>
<td>cachesize</td>
<td>smallint</td>
<td>not null Size of cache</td>
</tr>
<tr>
<td>storedby</td>
<td>char (8)</td>
<td>not null Authorization ID</td>
</tr>
<tr>
<td>timestamp</td>
<td>timestamp</td>
<td>NNWD Timestamp for row creation</td>
</tr>
<tr>
<td>plentries</td>
<td>smallint</td>
<td>NNWD Number of plan entries</td>
</tr>
</tbody>
</table>
### Table 26. EEEPLAN table (continued)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysentries</td>
<td>smallint</td>
<td>Number of system entries</td>
</tr>
<tr>
<td>expredicate</td>
<td>char (1)</td>
<td>Option for CURRENTDATA</td>
</tr>
<tr>
<td>deferprep</td>
<td>char (1)</td>
<td>Option for DEFER</td>
</tr>
<tr>
<td>currentserver</td>
<td>char (16)</td>
<td>Location name</td>
</tr>
<tr>
<td>degree</td>
<td>char (3)</td>
<td>Degree of parallelism</td>
</tr>
<tr>
<td>sqlrules</td>
<td>char (1)</td>
<td>SQL rules</td>
</tr>
<tr>
<td>disconnect</td>
<td>char (1)</td>
<td>Disconnect option</td>
</tr>
</tbody>
</table>

**The EEPATH table**

The EEPATH table is used to save selected PLAN_TABLE and DSN_STATEMNT_TABLE information that later is sent to SQL PA together with the extracted SQL statements.
### Table 27. EEEPATH table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queryno</td>
<td>Same definitions as the corresponding columns in PLAN_TABLE.</td>
<td>Same descriptions as the corresponding columns in PLAN_TABLE.</td>
</tr>
<tr>
<td>Qblockno</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applname</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planno</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tname</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accessype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mathcols</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accesscreator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accessname</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indexonly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortn_uniq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortn_join</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortn_orderby</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortn_groupby</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortc_uniq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortc_join</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortc_orderby</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortc_groupby</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tslockmode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prefetch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column fn_eval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixopseq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access_degree</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access_pgroup_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Join_degree</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Join_pgroup_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortc_pgroup_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sortn_pgroup_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallelism_mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merge_join_cols</td>
<td>Same definitions as the corresponding columns in DSN_STATEMENT_TABLE.</td>
<td>Same descriptions as the corresponding columns in DSN_STATEMENT_TABLE.</td>
</tr>
<tr>
<td>Correlation_name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page_range</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Join_type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qblock_type</td>
<td>Same definitions as the corresponding columns in PLAN_TABLE.</td>
<td>Same descriptions as the corresponding columns in PLAN_TABLE.</td>
</tr>
<tr>
<td>Primary_accesstype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table_type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stmt_type</td>
<td>Same definitions as the corresponding columns in DSN_STATEMENT_TABLE.</td>
<td>Same descriptions as the corresponding columns in DSN_STATEMENT_TABLE.</td>
</tr>
<tr>
<td>Cost_category</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procsu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mergc</td>
<td>Same definitions as the corresponding columns in PLAN_TABLE.</td>
<td>Same descriptions as the corresponding columns in PLAN_TABLE.</td>
</tr>
<tr>
<td>Mergn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### PLAN_TABLE contents

If you do not already have a PLAN_TABLE table, Easy Explain creates one if you have specified an SQL or QMF statement in the input stream to Easy Explain.
The PLAN_TABLE is located in the default database unless a row giving the names of the database and table space was inserted into the EEE_LOCATION view.

The contents of the PLAN_TABLE table is dependent on the DB2 version you are using. As shown in the following figure, the first 25 columns are created regardless of the DB2 release. Column numbers 26 through 28 were added by DB2 V2R2. Column numbers 29 through 30 were added by DB2 V2R3. Column numbers 31 through 34 were added by DB2 V3. Column numbers 35 through 43 were added by DB2 V4. Column numbers 44 through 46 were added by DB2 V5. Column numbers 47 through 49 were added by DB2 V6. Column numbers 50 and 51 were added by DB2 V7. And finally, column numbers 52 through 58 were added by DB2 V8.

Easy Explain compares the DB2 level and the columns available in the actual PLAN_TABLE, and automatically adds any missing columns.

The PLAN_TABLE table is normally prefixed by the primary user ID, unless a secondary SQL ID has been issued. In that case, the table is prefixed by the secondary SQL ID. For example, if the user FLORET has specified the following Easy Explain input:

```sql
//INPUT DD *
SET CURRENT SQLID = 'Group7'
QMF=SalesStat,owner=George
```

a PLAN_TABLE table is created under authorization ID Group7 if such a table does not exist.

**Note:** The creation of the PLAN_TABLE table and the addition of release-dependent columns only occurs if you have included an SQL or QMF statement in the input stream.

For a description of the individual columns, see the *DB2 SQL Reference*, (SC18-7426), under Explain statement.

<table>
<thead>
<tr>
<th>Column number</th>
<th>Column name</th>
<th>Data format</th>
<th>Null definition</th>
<th>Created in DB2</th>
<th>Included in tabular report</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QUERYNO</td>
<td>INTEGER</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>QBLOCKNO</td>
<td>SMALLINT</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>APPLNAME</td>
<td>CHAR (8)</td>
<td>NOT NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PROGNAME</td>
<td>CHAR (8)</td>
<td>NOT NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PLANNO</td>
<td>SMALLINT</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>METHOD</td>
<td>SMALLINT</td>
<td>NOT NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CREATOR</td>
<td>CHAR (8)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>TNAME</td>
<td>CHAR (18)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TABNO</td>
<td>SMALLINT</td>
<td>NOT NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ACCESSSTYPE</td>
<td>CHAR (2)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>MATCHCOLS</td>
<td>SMALLINT</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ACCESSCREATOR</td>
<td>CHAR (8)</td>
<td>NOT NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ACCESSNAME</td>
<td>CHAR (18)</td>
<td>NOT NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column number</td>
<td>Column name</td>
<td>Data format</td>
<td>Null definition</td>
<td>Created in DB2</td>
<td>Included in tabular report</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>14</td>
<td>INDEXONLY</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SORTN_UNIQ</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>SORTN_JOIN</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>SORTN_ORDERBY</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>SORTN_GROUPBY</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SORTC_UNIQ</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>SORTC_JOIN</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>SORTC_ORDERBY</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>SORTC_GROUPBY</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>TSOLOCKMODE</td>
<td>CHAR (3)</td>
<td>NOT NULL</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>TIMESTAMP</td>
<td>CHAR (16)</td>
<td>NOT NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>REMARKS</td>
<td>VARCHAR(254)</td>
<td>NOT NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>PREFETCH</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>V2R2</td>
<td>Y</td>
</tr>
<tr>
<td>27</td>
<td>COLUMN_FN_EVAL</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>V2R2</td>
<td>Y</td>
</tr>
<tr>
<td>28</td>
<td>MIXOPSEQ</td>
<td>SMALLINT</td>
<td>NOT NULL</td>
<td>V2R2</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>VERSION</td>
<td>VARCHAR(64)</td>
<td>NOT NULL</td>
<td>V2R3</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>COLLID</td>
<td>CHAR (18)</td>
<td>NOT NULL</td>
<td>V2R3</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>ACCESS_DEGREE</td>
<td>SMALLINT</td>
<td>Nullable</td>
<td>V3</td>
<td>Y</td>
</tr>
<tr>
<td>32</td>
<td>ACCESS_PGROUP_ID</td>
<td>SMALLINT</td>
<td>Nullable</td>
<td>V3</td>
<td>Y</td>
</tr>
<tr>
<td>33</td>
<td>JOIN_DEGREE</td>
<td>SMALLINT</td>
<td>Nullable</td>
<td>V3</td>
<td>Y</td>
</tr>
<tr>
<td>34</td>
<td>JOIN_PGROUP_ID</td>
<td>SMALLINT</td>
<td>Nullable</td>
<td>V3</td>
<td>Y</td>
</tr>
<tr>
<td>35</td>
<td>SORTC_PGROUP_ID</td>
<td>SMALLINT</td>
<td>Nullable</td>
<td>V4R1</td>
<td>Y</td>
</tr>
<tr>
<td>36</td>
<td>SORTN_PGROUP_ID</td>
<td>SMALLINT</td>
<td>Nullable</td>
<td>V4R1</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>PARALLELISM_MODE</td>
<td>CHAR (1)</td>
<td>Nullable</td>
<td>V4R1</td>
<td>Y</td>
</tr>
<tr>
<td>38</td>
<td>MERGE_JOIN_COLS</td>
<td>SMALLINT</td>
<td>Nullable</td>
<td>V4R1</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>CORRELATION_NAME</td>
<td>CHAR (18)</td>
<td>Nullable</td>
<td>V4R1</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>PAGE_RANGE</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>V4R1</td>
<td>Y</td>
</tr>
<tr>
<td>41</td>
<td>JOIN_TYPE</td>
<td>CHAR (1)</td>
<td>NOT NULL</td>
<td>V4R1</td>
<td></td>
</tr>
</tbody>
</table>
### Table 28. PLAN_TABLE DB2 dependencies (continued)

<table>
<thead>
<tr>
<th>Column number</th>
<th>Column name</th>
<th>Data format</th>
<th>Null definition</th>
<th>Created in DB2</th>
<th>Included in tabular report</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>GROUP_MEMBER</td>
<td>CHAR (8)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V4R1</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>IBM_SERVICE_DATA</td>
<td>VARCHAR(254)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V4R1</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>WHEN_OPTIMIZE</td>
<td>CHAR (1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V5R1</td>
<td>Y</td>
</tr>
<tr>
<td>45</td>
<td>QBLOCK_TYPE</td>
<td>CHAR (6)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V5R1</td>
<td>Y</td>
</tr>
<tr>
<td>46</td>
<td>BIND_TIME</td>
<td>TIMESTAMP</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V5R1</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>OPTHINT</td>
<td>CHAR (8)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V6R1</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>HINT_USED</td>
<td>CHAR (8)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V6R1</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>PRIMARY_ACCESSTYPE</td>
<td>CHAR (1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V6R1</td>
<td>Y</td>
</tr>
<tr>
<td>50</td>
<td>PARENT_QBLOCKNO</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V7R1</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>TABLE_TYPE</td>
<td>CHAR (1)</td>
<td>Nullable</td>
<td>V7R1</td>
<td>Y</td>
</tr>
<tr>
<td>52</td>
<td>TABLE_ENCODE</td>
<td>CHAR (1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V8R1</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>TABLE_SCCSID</td>
<td>CHAR (1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V8R1</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>TABLE_MCCSID</td>
<td>CHAR (1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V8R1</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>TABLE_DCCSID</td>
<td>CHAR (1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V8R1</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>ROUTINE_ID</td>
<td>INTEGER</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V8R1</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>CTEREF</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V8R1</td>
<td></td>
</tr>
<tr>
<td>Column number</td>
<td>Column name</td>
<td>Data format</td>
<td>Null definition</td>
<td>Created in DB2</td>
<td>Included in tabular report</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>58</td>
<td>STMTTOKEN</td>
<td>VARCHAR(240)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V8R1</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>PARENT_PLANNO</td>
<td>SMALLINT</td>
<td>NOT NULL</td>
<td>V9R1</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>BIND_EXPLAIN_ONLY</td>
<td>CHAR(1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V10R1</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>SECTNOI</td>
<td>INTEGER</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V10R1</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>EXPLAIN_TIME</td>
<td>TIMESTAMP</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V10R1</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>MERGC</td>
<td>CHAR(1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V10R1</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>MERGN</td>
<td>CHAR(1)</td>
<td>NOT NULL WITH DEFAULT</td>
<td>V10R1</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 11. Messages and return codes

These topics describe the error messages and return codes that are issued by DB2 SQL Performance Analyzer.

DB2 SQL Performance Analyzer messages and return codes

DB2 SQL Performance Analyzer messages are grouped by the following:

- Output report messages
- Statistics collector messages
- DB2 SQL Performance Analyzer ISPF panel messages

The warnings, alerts, recommendations, guidelines, notes and good news messages from the Explain and Detailed reports are also listed.

DB2 SQL Performance Analyzer messages are of the form ANLnnnnt, where:

- \( nnnn \) is the message number.
- \( t \) is a one-character suffix with the value I for informational messages, W for warnings, and E for errors.

Return codes

A return code is set for most DB2 SQL Performance Analyzer messages using the conventions shown in Table 29.

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Type of Message</th>
<th>ANLSQLPA</th>
<th>ANLQMF</th>
<th>ANLPRCR and ANLPRER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANL0000I</td>
<td>Informational</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ANL7xxxI</td>
<td>Guidelines and Good News</td>
<td>0</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>ANL6xxxI</td>
<td>Notes® and Recommendations</td>
<td>0</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>ANL5xxxI</td>
<td>Default stats, parameters, autofixes</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ANL5xxxW</td>
<td>Warning messages</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>ANL5xxxE</td>
<td>Error messages</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

In the message IDs, \( n = 1, 2, 3, \) or 4.

Output report messages

The following messages are issued from ANLSQLPA and ANLQMF.

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANL0000I</td>
<td>PLAN_TABLE WILL BE USED TO HOUSE EXPLAIN OUTPUT FOR THIS RUN</td>
<td>Informational. informs you of the plan table name used for evaluations.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>None.</td>
<td></td>
</tr>
</tbody>
</table>
ANL0100I  ACCORDING TO THE DB2-CATALOG, YOU DO NOT HAVE A xxx_TABLE TABLE. A xxx_TABLE TABLE WILL NOW BE CREATED FOR AUTHORIZATION-ID: authid

Explanation:  Easy Explain was not able to find an xxx_TABLE for the current authorization ID. Easy Explain creates an xxx_TABLE in the default database (SYSDEFLT), providing that proper authorization exists. The xxx_TABLE can be a PLAN_TABLE, a DSN_STATEMNT_TABLE, or a DSN_FUNCTION_TABLE. This message can be followed by message ANL0102I or by message ANL0600E.

User response:  None.

ANL0101I  COLUMN column name HAS BEEN ADDED TO authid.EEEPATH

Explanation:  Easy Explain has found that you have specified a dynamic explain request using an existing EEEPATH using an older SQL PA version. Easy Explain is upgrading the EEEPATH by adding the appropriate columns to the existing EEEPATH.

User response:  None.

ANL0102I  xxx_TABLE NAMED authid.xxx_TABLE CREATED IN THE DEFAULT DATABASE

Explanation:  Easy Explain has created an xxx_TABLE for authorization ID authid in the default database (SYSDEFLT). The xxx_TABLE can be a PLAN_TABLE, a DSN_STATEMNT_TABLE, or a DSN_FUNCTION_TABLE. This message is preceded by message ANL0100I.

User response:  None.

ANL0104I  COLUMN column name HAS BEEN ADDED TO authid.PLAN_TABLE

Explanation:  Easy Explain has found that you have specified a dynamic explain request using an existing PLAN_TABLE using an older DB2 version. Easy Explain is upgrading the PLAN_TABLE by adding the appropriate columns to the existing PLAN_TABLE.

User response:  None.

ANL0106I  COLUMN column name HAS BEEN ADDED TO authid.DSN_STATEMNT_TABLE

Explanation:  Easy Explain has found that you have specified a dynamic explain request using an existing DSN_STATEMNT_TABLE using an older DB2 version. Easy Explain is upgrading the DSN_STATEMNT_TABLE by adding the appropriate columns to the existing DSN_STATEMNT_TABLE.

User response:  None.

ANL0108I  COLUMN column name HAS BEEN ADDED TO authid.DSN_FUNCTION_TABLE

Explanation:  Easy Explain has found that you have specified a dynamic explain request using an existing DSN_FUNCTION_TABLE using an older DB2 version. Easy Explain is upgrading the DSN_FUNCTION

_TABLE by adding the appropriate columns to the existing DSN_FUNCTION _TABLE.

User response:  None.

ANL0110I  COLUMN column name HAS BEEN ADDED TO authid.EEEPATH

Explanation:  Easy Explain has found that you have specified a dynamic explain request using an existing EEEPATH using an older SQL PA version. Easy Explain is upgrading the EEEPATH by adding the appropriate columns to the existing EEEPATH.

User response:  None.

ANL0116I  THE LOCATION FOR xxx_TABLE IS SPECIFIED AS xxxxx.xyyyyyy

Explanation:  The installation has specified that xxx_TABLE tables are created in the database named xxxxx, and within that database in the table space named xyyyyyy. Easy Explain attempts to create the xxx_TABLE table in that location. The xxx_TABLE can be a PLAN_TABLE, a DSN_STATEMNT_TABLE, or a DSN_FUNCTION_TABLE.

User response:  None.

ANL0118I  NO LOCATION SPECIFIED FOR xxx_TABLE. WILL USE DEFAULT DATABASE

Explanation:  The installation has not specified a location (database and table space names) in the EEE_LOCATION table for xxx_TABLE tables to be created. Consequently, Easy Explain creates the xxx_TABLE table in the DB2 default database. The xxx_TABLE can be a PLAN_TABLE, a DSN_STATEMNT_TABLE, or a DSN_FUNCTION_TABLE.

User response:  None.

ANL0120I  INDEX INFORMATION SUPPRESSED BY USER REQUEST

Explanation:  You have specified INDEX=N for the PLN, QMF, or QNO keyword. This means that no column information for the chosen index is shown.

User response:  None.

ANL0130I  NO STATISTICS AVAILABLE - EXECUTE RUNSTATS

Explanation:  Easy Explain has found that no DB2 catalog statistics are available for the current table space. Before starting Easy Explain again, run RUNSTATS against the table space.

User response:  None.
**ANL0136I**  HINT-ID (xxx) WAS ADDED FOR STATEMENT yyy IN PROGRAM zzz

**Explanation:** Easy Explain has added the specified hint ID to the specified row (statement number yyy for program zzz) in the PLAN_TABLE table.

**User response:** None.

**ANL0140I**  NOTE: dbname.tsname CONTAINS xxx ACTIVE PAGES - CONSIDER A USABLE INDEX

**Explanation:** This message is produced if the DB2 Optimizer has chosen a table space scan for a table located in a table space larger than 10 active pages. If you want the access path to be INDEX-based, create a usable index.

**User response:** None.

**ANL0142E**  TABLE authid.tname NOT FOUND IN SYSIBM.SYSTABLES - SQL-CODE: -xxx

**Explanation:** Easy Explain was not able to find the named table in SYSIBM.SYSTABLES. The table might be a result of a materialization. Otherwise check the non-zero SQL code (-xxx) and follow the instructions given in the DB2 Messages and Codes manual.

**User response:** None.

**ANL0144I**  ALTER TABLE ... UPDATED TABLE authid.PLAN_TABLE TO V8 FORMAT

**Explanation:** Easy Explain has updated the column definitions for the PLAN_TABLE table, so it reflects the expanded DB2 Version 8 column definitions.

**User response:** None.

**ANL0146I**  ALTER TABLE ... UPDATED TABLE authid.DSN_STATEMNT_TABLE TO V8 FORMAT

**Explanation:** Easy Explain has updated the column definitions for the DSN_STATEMNT_TABLE table, so it reflects the expanded DB2 Version 8 column definitions.

**User response:** None.

**ANL0148I**  ALTER TABLE ... UPDATED TABLE authid.DSN_FUNCTION_TABLE TO V8 FORMAT

**Explanation:** Easy Explain has updated the column definitions for the DSN_FUNCTION_TABLE table, so it reflects the expanded DB2 Version 8 column definitions.

**User response:** None.

**ANL0150I**  TABLE xxx.xxxxxx IS AN IMMEDIATE TABLE - NO DATA AVAILABLE

**Explanation:** Easy Explain has decided that the table being investigated is an immediate temporary table. Easy Explain skips processing the table and continues with other work.

**User response:** None.

**ANL0160I**  AUTHORIZATION TABLE DOES NOT EXIST - WILL USE DEFAULT AUTHORIZATIONS

**Explanation:** The user authorization table, binder.EEE_EEAUTH, could not be found. Easy Explain uses default authorization values (messages shown in NLS if available, SQL statements are formatted, plan history is not shown, remote packages explained). Operation continues.

**User response:** None.

**ANL0162I**  PACKAGE REPORTS NOT REQUESTED

**Explanation:** You have specified that packages for a specific plan not be explained. Easy Explain explains all DBRMs, but no packages. Operation continues.

**User response:** None.

**ANL0164I**  COLLECTION SPECIFICATION xxxxxxx.* HAS MORE THAN 10 VERSIONS USE PACKAGE=collid.name.(version) TO GET A DETAILED LISTING

**Explanation:** You have specified an asterisk (*) as the package name for collection (xxxxxxx). Easy Explain has found that the collection contains more than 10 packages. To explain all packages in the collection, set FORCE=YES on the package statement. Operation continues.

**User response:** None.

**ANL0166I**  PACKAGE xxxxxx IN COLLECTION yyyyyyy HAS MORE THAN 5 VERSIONS USE PACKAGE=collid.name.(version) TO GET A DETAILED LISTING

**Explanation:** You have specified package name (xxxxxx) for collection (yyyyyyyy). Easy Explain has found that the package exists in more than five versions. The version identifications are listed. To explain all versions of the package, set FORCE=YES on the package statement. Operation continues.

**User response:** None.
ANL0168I  THE PACKAGE NAME IS SPECIFIED AS A THREE-PART NAME. LOCATION IGNORED.

Explanation: You have specified the package name with a location name in front of the collection and package identifications. The Package (PKG) statement, however, does not support specification of a location name. The location name is ignored, and the operation continues.

User response: None.

ANL0170I  DD-NAME FOR INPUT WAS SPECIFIED AS CARD - SHOULD BE INPUT

Explanation: The ddname for the input data set was in the previous Easy Explain version named CARD. In the current version of Easy Explain, the ddname is INPUT. For compatibility reasons the name CARD can still be used, but change it to INPUT as soon as possible.

User response: None.

ANL0172I  DD-NAME FOR OUTPUT (REPORT) WAS MISSING - WILL USE SYSTSPRT

Explanation: The ddname for the print data set REPORT was not specified. Easy Explain instead uses the ddname SYSTSPRT, when printing the Easy Explain report. Before running again, add the following job control card: //REPORT DD SYSOUT=*  

User response: None.

ANL0174I  DD-NAME FOR OUTPUT (SYSTSPRT) WAS MISSING

Explanation: The ddname for the print data set SYSTSPRT was not specified. If tracing was enabled, the trace facility is now disabled. Operation continues. Before running again, add the following job control card: //SYSTSPRT DD SYSOUT=*  

User response: None.

ANL0176I  DD-NAME FOR OUTPUT DATA SET (SQLPA) WAS MISSING

Explanation: The ddname for the output data set SQLPA was not specified. To have Easy Explain produce output meant for SQL Performance Analyzer, disable this process. Before running again, add the following job control card: //SQLPA DD DSN=xxx,DISP=yy  

User response: None.

ANL0180I  CONNECTION TO SERVER xxxxx SUCCESSFUL SERVER TYPE IS mnn VER. r REL. p MOD. q

Explanation: An SQL Connect statement was issued by Easy Explain to connect to the server location specified by the LOCATION=xxxx parameter. The connection was established. mnn is the type of the server; valid types are: DB2, DB2/VM, DB2/400, DB2/2, and DB2/6000. r, p, and q gives the version, release, and modification level of the server system.

User response: None.

ANL0182I  CONNECT RESET TO xxxxx SUCCESSFUL

Explanation: An SQL Connect Reset statement was issued by Easy Explain to reconnect to the server system where Easy Explain is run. The connection was re-established.

User response: None.

ANL0184I  CURRENT SERVER NAME COULD NOT BE ACQUIRED - WILL USE BLANKS

Explanation: An SQL Set Current Server statement was issued by Easy Explain to set the new server, but the call failed. Easy Explain continues running.

User response: None.

ANL0300W  INVALID INPUT PARAMETER: FIRST 45 CHARACTERS OF INPUT SPEC.

Explanation: An input record with a keyword different from LVL (or Level), QNO (or Queryno), PLN (or Plan), PKG (or Package), QMF, or SQL has been received by Easy Explain. The first 45 characters of the input record are shown. Easy Explain proceeds to the next input record.

User response: None.

ANL0302W  OWNER SUBPARAMETER (authid) INVALID - PARAMETER IGNORED

Explanation: An input record with the QNO keyword has an incorrect specified authorization ID following the OWNER= subparameter. Easy Explain ignores the specified authorization ID and continue processing using the PLAN_TABLE of the job submitter or TSO user.

User response: Specify a valid authorization ID for QNO keyword, and issue the request again.
Explanation: An input record with the QNO (or QueryNo), PLN (or Plan), PKG (Package), or QMF keyword has an incorrect specified value following the INDEX= subparameter. Valid values are strings starting with Y, N, or T. Easy Explain continues processing with a default value of Y.

User response: Specify a valid QNO, PLN, PKG, or QMF value, and issue the request again.

---

**ANL0306W** SQL-STATEMENT WAS NOT TERMINATED BY AN ';' - STATEMENT IGNORED

Explanation: The statement following the SQL keyword did not end with a semicolon (;). You can specify the SQL statement on one line or on several lines, but it must always end with semicolon. The statement is ignored and Easy Explain proceeds to the next input record.

User response: End the statement with a semicolon, and reissue the request.

---

**ANL0308W** INVALID QNO FORMAT: FIRST 50 CHARACTERS OF INPUT SPE.

Explanation: The parameter following the QNO= or QueryNo= keyword is not numeric or the number does not immediately follow the equal sign. The first 50 characters of the statement is shown in the warning. The statement is ignored and Easy Explain proceeds to the next input record.

User response: Specify a numeric value for the QNO keyword immediately following the equal sign, and issue the request again.

---

**ANL0310W** QUERY NO. xxxx DOES NOT EXIST FOR USER authid - REQUEST IGNORED

Explanation: The specified query number (xxxx) following the QNO= or QueryNo= keyword could not be located in the accessed PLAN_TABLE. The statement is ignored and Easy Explain proceeds to the next input record.

User response: Verify that the query number does exist in the table. Also verify that the correct PLAN_TABLE is being accessed. If necessary specify the correct authorization ID in the owner parameter. Specify a valid query number, and reissue the request.

---

**ANL0314W** QNO. xxx IN DBRM yyyyyyy, PLAN zzzzzzz NOT IN authid PLAN_TABLE

Explanation: Easy Explain was not able to locate query number (xxx) for plan (zzzzzzz) and DBRM (yyyyyy) in the PLAN_TABLE of the plan binder. Easy Explain continues looking for the next query number for plan (zzzzzzz).

User response: Specify a valid query number for the specified plan and a valid DBRM for the PLAN_TABLE, and issue the request again.

---

**ANL0316W** QNO. xxx IN PACKAGE yyyyyyy, zzzzzzzz NOT IN authid PLAN_TABLE

Explanation: Easy Explain was not able to locate query number (xxx) for collection ID (yyyyyy) and package name (zzzzzzz) in the PLAN_TABLE of the plan binder. Easy Explain continues looking for the next query number for collection ID (yyyyyy) and package name (zzzzzzz).

User response: Specify a valid query number for the specified collection ID and a valid package name for the PLAN_TABLE, and issue the request again.

---

**ANL0320W** PLAN xxxxxxx NOT BOUND WITH EXPLAIN OPTION - DYNAMIC EXPLAIN ISSUED

Explanation: The plan specified after the PLN= or Plan= keyword was not bound with the Explain option, but the rows in the SYSIBM.SYSSTMT table have been deleted for the given DBRM. The statement is ignored and Easy Explain proceeds to the next input record.

User response: Rebind the plan, and run the job again with the same input record.

---

**ANL0322W** DBRM xxxxxxx HAS NO ROWS IN SYSIBM.SYSSTMT - REQUEST IGNORED

Explanation: The plan specified after the PLN= or Plan= keyword was bound with the Explain option, but the rows in the SYSIBM.SYSSTMT table have been deleted for the given DBRM. The statement is ignored and Easy Explain proceeds to the next input record.

User response: Rebind the plan, and run the job again with the same input record.

---

**ANL0324W** PLAN_TABLE FOR USER authid DOES NOT EXIST - REQUEST IGNORED

Explanation: The authorization ID given as OWNER subparameter for the QNO keyword does not own a PLAN_TABLE. The statement is ignored and Easy Explain proceeds to the next input record.

User response: Specify a valid authorized ID for the QNO keyword, and issue the request again.
ANL0330W • ANL0344W

ANL0330W SUBPARAMETER FIRST (xxxx) INVALID - PARAMETER IGNORED
Explanation: An input record with the PLN (or PLAN) keyword has an incorrect specified value following the FIRST= subparameter. Easy Explain ignores the specified value and continues processing using a default value of 1.
User response: Specify a valid value for the PLN keyword, and reissue the request.

ANL0332W SUBPARAMETER LAST (xxxx) INVALID - PARAMETER IGNORED
Explanation: An input record with the PLN (or PLAN) keyword has an incorrect specified value following the LAST= subparameter. Easy Explain ignores the specified value and continues processing using a default value of 99999999.
User response: Specify a valid value for the PLN keyword, and reissue the request.

ANL0334W ACCTYPE SUBPARAMETER (xxx) INVALID - DEFAULT VALUE USED
Explanation: An input record with the PLN (or PLAN) keyword has an incorrect specified value xxx following the ACCTYPE= subparameter. Easy Explain ignores the specified value and continues processing using a default value of All.
User response: Specify a valid value for the PLN keyword, and reissue the request.

ANL0336W LEVEL SUBPARAMETER (string) INVALID - CURRENT VALUE USED (value)
Explanation: An input record with the PLN (or PLAN) or PKG (or PACKAGE) keyword has an incorrect specified value string following the LEVEL= subparameter. Easy Explain ignores the specified value and continues processing using the current level value of value, where value is one of the following options:
- DET
- SUM
- NOC

User response: Specify a valid value for the PLN keyword, and reissue the request.

ANL0340W SUBPARAMETER TABLE (xxx) INVALID - PARAMETER IGNORED
Explanation: An input record with the PLN (or PLAN) keyword has an incorrect specified value xxx following the TABLE= subparameter. Easy Explain ignores the specified value and continues processing using all tables within the plan.
User response: Specify a valid value for the PLN keyword, and reissue the request.

ANL0342W TABLE NAME xxx DOES NOT EXIST - PARAMETER IGNORED
Explanation: An input record with the PLN (or PLAN) keyword has an incorrect specified value xxx following the TABLE= subparameter. Easy Explain ignores the specified value and continues processing using all tables within the plan.
User response: Specify a valid value for the PLN keyword, and reissue the request.

ANL0344W SUBPARAMETER LOCATION (xxx) INVALID - PARAMETER IGNORED
Explanation: An input record with the PLAN or PACKAGE keyword has an incorrect specified value xxx for the LOCATION= subparameter. Easy Explain ignores the specified value and continues processing using the last specified location name, or the home server if no location value has been specified.
User response: Specify a valid value for the PLAN or
PACKAGE keyword, and issue the request again.

**ANL0358W**  
**STMT.NO. xxxx IN yyyy.zz**  
**USES UNQUALIFIED TABLES NAME(S)**

**Explanation:** A dynamic Explain has been issued for a PLAN or PACKAGE. However, the implicit qualifier yyyy for the unqualified tables in plan or package zzzzz is not the same as the Easy Explain submitter. Easy Explain is not able to substitute the creator ID for the tables used in the statements, and is consequently ignoring the SQL statement.

**User response:** Remove the PLAN or PACKAGE, or run the job with the valid authorization ID.

**ANL0360W**  
**PACKAGE STATEMENT IS NOT SUPPORTED IN DB2 REL. x.y - REQUEST IGNORED**

**Explanation:** An input record with the PKG (or Package) keyword has been read by Easy Explain, but Easy Explain is running on a DB2 2.1 or 2.2 subsystem, where the package concept is unknown. The statement is ignored and Easy Explain proceeds to the next input record.

**User response:** None.

**ANL0362W**  
**INVALID COLLECTION AND PACKAGE SPECIFICATION: xxxxxxx**

**Explanation:** An input record with the PKG (or Package) keyword has an incorrect specified collection and package name combination. Easy Explain ignores the specified value and continues processing the next input statement.

**User response:** Specify a valid PKG and collection name combination, and issue the request again.

**ANL0364W**  
**INVALID PACKAGE NAME: xxxxxxx**

**Explanation:** An input record with the PKG (or Package) keyword has an incorrect specified package name. Easy Explain ignores the specified value and continues processing the next input statement.

**User response:** Specify a valid PKG keyword, and issue the request again.

**ANL0366W**  
**NO LOCAL PACKAGE LIST FOUND FOR PLAN xxxx**

**Explanation:** The PLENTRIES column in the SYSIBM.SYSPLAN table has indicated that at least one package list is associated with the plan. Easy Explain, however, is not able to find a row in SYSIBM.SYSPACKLIST for the specified plan (xxxxx). The statement is ignored and Easy Explain proceeds to the next input statement.

**User response:** Specify a valid package list for the specified plan, and issue the request again.

**ANL0368W**  
**LOCATION CANNOT BE DETERMINED UNTIL EXECUTION TIME - PACKAGE IGNORED**

**Explanation:** The LOCATION column in the SYSIBM.SYSPACKLIST table contains an asterisk (*) for the specified plan name and the given sequence number. This means that Easy Explain has no way of knowing on what server location the package might be found. The package is ignored and Easy Explain proceeds to the next package in the package list.

**User response:** Specify a valid LOCATION name for the specified plan and sequence number, and issue the request again.

**ANL0370W**  
**PACKAGE xxxxxxx.yyyyy.zz**  
**NOT BOUND WITH EXPLAIN OPTION - DYNAMIC EXPLAIN ISSUED**

**Explanation:** The package specified as xxxxxxx.yyyyy.zz after the PKG= or PACKAGE= keyword was not bound with the EXPLAIN option. Easy Explain is performing a dynamic explain of the statements in the package, and shows the result of this dynamic explain.

**User response:** None.

**ANL0372W**  
**THE STATEMENT xxxxxxxx WILL RESULT IN y PACKAGES WITH A TOTAL OF z SQL STATEMENTS (INCLUDING NON-EXPLAINABLE). IF YOU WANT TO EXPLAIN THAT AMOUNT OF PACKAGES, YOU SHOULD SPECIFY xxxxx,FORCE=YES**

**Explanation:** Easy Explain has found that the package statement that you have specified results in an explain of more than one package with a total of more than 300 SQL statements (counting both explainable and non-explainable). The statement is ignored and Easy Explain proceeds to the next input statement.

**User response:** To explain the number of packages specified, reissue the request specifying the FORCE=YES subparameter.

**ANL0374W**  
**NO PACKAGES FOUND FOR: xxxxxxxxxxxxx**

**Explanation:** Easy Explain was not able to find any packages in SYSIBM.SYSPACKAGE that would satisfy the input specification. The statement is ignored and Easy Explain proceeds to the next input statement.

**User response:** Specify a valid PACKAGE, and issue the request again.
ANL0376W  NO PACKAGES FOUND FOR PLAN:
xxxxxxx - PACKAGE: yyyyyyy
LOCATION: zzzzzz - COLL.ID: qqqqqqq

Explanation: Easy Explain was not able to find any packages in SYSIBM.SYSPACKAGE for plan xxxxxx. The statement is ignored and Easy Explain proceeds to the next input statement.

User response: Specify a valid PACKAGE for the specified plan, and issue the request again.

ANL0377W  PACKAGE xxxxxx.yyyyyyy.(zzzzzz) HAS NO ROWS IN SYSIBM.SYSPACKSTMT

Explanation: Easy Explain was not able to find any rows in SYSIBM.SYSPACKSTMT for the specified package in the specified collection. The statement is ignored and Easy Explain proceeds to the next input statement.

User response: Specify a valid PACKAGE for the specified collection, and issue the request again.

ANL0378W  QMF QUERY authid.gname DOES NOT EXIST - REQUEST IGNORED

Explanation: A QMF query name specified after the QMF= keyword does not exist. If the Easy Explain submitter is not the owner of the query, the query name must be prefixed with the authorization ID, for example, authid.gname. The statement is ignored and Easy Explain proceeds to the next input statement.

User response: Verify that the query name is spelled correctly, and issue the request again.

ANL0380W  QMF OBJECT authid.gname IS A xxxx AND NOT A QUERY - REQUEST IGNORED

Explanation: The QMF object specified as the query name is not a query, but rather a xxxx, where xxxx is a PROC or a FORM. The statement is ignored and Easy Explain proceeds to the next input statement.

User response: Specify a valid query name, and issue the request again.

ANL0382W  QUERY authid.gname IS A xxxx QUERY, MUST BE SQL - REQUEST IGNORED

Explanation: The QMF query specified as the query name is not an SQL query, but rather a xxxx, where xxxx is QBE or PROMPTED. The statement is ignored and Easy Explain proceeds to the next input statement.

User response: If the QMF version is at least Version 2 Release 3, use the CONVERT command to transform the query into an SQL query.

ANL0384W  QMF QUERY authid.gname IS RESTRICTED TO OWNER - REQUEST IGNORED

Explanation: The QMF query specified as the query name can only be explained by its owner. The statement is ignored and Easy Explain proceeds to the next input record.

User response: To explain a QMF query created by a different authorization ID, save the query with the SHARE=YES parameter.

ANL0388W  END OF QMF QUERY NOT FOUND - REQUEST IGNORED

Explanation: The QMF statement was found to be larger than 32 K bytes. Either the QMF query is more than 409 lines or the end-of-statement marker was not found. In either case, Easy Explain ignores the request and proceeds to the next input record.

User response: Verify that the QMF statement is less then 32 K bytes, less than 409 lines, and that it contains an end-of-statement marker, and issue the request again.

ANL0394W  INDEX INFORMATION NOT INSERTED IN EEEWORK TABLE - SQL-CODE: -xxx

Explanation: You requested a condensed index listing by specifying,INDEX=Cond, on the PLAN or PACKAGE statement. Easy Explain was not able to save part of the index information in the EEEWORK table. The index parameter is ignored and Easy Explain continues processing.

User response: None.

ANL0396W  OPEN OF C_WORK FAILED - SQL-CODE: -xxx

Explanation: You requested a condensed index listing by specifying,INDEX=Cond, on the plan or package statement. Easy Explain was not able to select the index information for the specified plan or package in the EEEWORK table. The index parameter is ignored and Easy Explain continues processing.

User response: None.

ANL0398W  OPEN OF C_WORK2 FAILED - SQL-CODE: -xxx

Explanation: You requested a condensed index listing by specifying,INDEX=Cond, on the PLAN or PACKAGE statement. Easy Explain was not able to select the index information for the specified plan or package in the EEEWORK table. The index parameter is ignored and Easy Explain continues processing.

User response: None.
ANL0420W LENGTH OF SQL STRING INVALID:
xxxxx - DEFAULT OF 79 USED
Explanation: You specified a line length of the printed SQL statement in the report. The length is invalid and has been set to 79 (default value). Easy Explain ignores the specified value and continues processing.
User response: Specify a larger valid line length, and issue the request again.

ANL0422W FORMAT OF SQL STRING INVALID:
xxxxx - DEFAULT OF 79 USED
Explanation: You specified a line length of the printed SQL statement in the report. The format is invalid and the value has been set to 79 (default value). Easy Explain ignores the specified value and continues processing.
User response: Specify a valid line length in the correct format, and issue the request again.

ANL0424W LENGTH OF SQL STRING OUT OF RANGE:
xxxxx - DEFAULT OF 79 USED
Explanation: You specified a line length of the printed SQL statement in the report. The value was outside the possible range (40 to 79). Easy Explain ignores the specified value and continues processing using the default value of 79.
User response: Specify a line length within the valid range (40 to 79), and reissue the request.

ANL0426W QUERYNO xxxx IS NOT NUMERIC - REQUEST IGNORED
Explanation: You specified a non-numeric QUERYNO as xxxx. The QUERYNO must be specified as a numeric number. Easy Explain ignores the specified value and continues processing using the default value of 79.
User response: Specify a numeric QUERYNO, and issue the request again.

ANL0450W INVALID HINT COMMAND:
xxxxxxxxxxxxxxxx
Explanation: You specified a SET CURRENT OPTIMIZATION HINT command that is in error. The first 50 bytes of the command is displayed. Easy Explain ignores the specified value and continues processing using the default value of 79.
User response: Specify a valid SET CURRENT OPTIMIZATION HINT command, and issue the request again.

ANL0452W HINT-ID NUMBER NOT FOUND IN EEEHINT TABLE - REQUEST IGNORED
Explanation: The hint ID number specified could not be found in the EEEHINT table. The statement is ignored and Easy Explain proceeds to the next input statement.
User response: Specify a hint ID that is in the EEEHINT table, and issue the request again.

ANL0454W HINT-ID WAS NOT ADDED TO THE PLAN_TABLE - REQUEST IGNORED
Explanation: Easy Explain tried to update your PLAN_TABLE table with the specified hint ID, but the operation failed. The statement is ignored and Easy Explain proceeds to the next input statement.
User response: None.

ANL0456W HOST VARIABLE INFORMATION COULD NOT BE GATHERED
Explanation: Easy Explain tried to gather host variable information, but the operation failed. The host variable information is ignored and Easy Explain processing proceeds.
User response: None.

ANL0458W Package package_name must be bound before it can be used.
Explanation: A package that was bound by using DB2 V9.1 or an earlier release was explained by using a DB2 11 subsystem.
User response: Rebind the package by using a DB2 11 subsystem.

ANL0500E PREPARE OF S_DYNx FAILED - SQL-CODE: -xxx
Explanation: A prepare statement failed during preparing a select statement for S_DYNx, where x is 1, 2, 3, 4, or 5. A formatted print of the SQLCA and an error message is provided.
User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.
Severity: 8

ANL0504E DECLARE OF S_DYNx FAILED - SQL-code: -xxx
Explanation: A declare statement failed during declaring the cursor C_PLANx for S_DYNx, where x is 1, 2, 3, 4, or 5. A formatted print of the SQLCA and an error message is provided.
User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0508E  OPEN OF C_PLANx FAILED, SQL-CODE: -xxx

Explanation: An open statement failed during open of the cursor C_PLANx for S_DYNx, where x is 1, 2, 3, 4, or 5. A formatted print of the SQLCA and an error message is provided.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0510E  RESETTING OF PRIMARY SQLID (auth.id) FAILED, WILL USE: auth.id

Explanation: Easy Explain failed to set the primary SQL ID (primary authorization ID). The program continues using the logon user ID as the authorization ID. A formatted print of the SQLCA and an error message is provided. This message is only produced if running under DB2 Version 2.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0512E  SETTING OF SECONDARY SQLID (auth.id) FAILED, WILL USE: auth.id

Explanation: Easy Explain failed to set the secondary SQL ID (secondary authorization ID). The program continues using the primary authorization ID. A formatted print of the SQLCA and an error message is provided. This message is only produced if running under DB2 Version 2.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0514E  CURRENT SQLID COULD NOT BE OBTAINED - WILL USE AUTHORIZATION-ID: auth.id

Explanation: Easy Explain failed to select the current SQL ID (secondary authorization ID). The program continues using the primary authorization ID. A formatted print of the SQLCA and an error message is provided. This message is only produced if running under DB2 Version 2.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0520E  OPEN OF C_KEYS FAILED, SQL-CODE: -xxx

Explanation: An error was detected during opening a cursor for SYSIBM.SYSKEYS joined with SYSIBM.SYSINDEXES. If the SQL code is -911 or -913, Easy Explain waits a short period and then tries to reprocess the current statement in the plan. Refer to message ANL0532E. For other SQL codes, Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

User response: Reissue the request when the resource is available.

Severity: 8

ANL0522E  FETCH FROM C_KEYS FAILED, SQL-CODE: -xxx

Explanation: An error was detected during fetch from SYSIBM.SYSKEYS joined with SYSIBM.SYSINDEXES. If the SQL code is -911 or -913, Easy Explain waits a short period and then tries to reprocess the current statement in the plan. Refer to message ANL0532E. For other SQL codes, Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

User response: Reissue the request when the resource is available.

Severity: 8

ANL0524E  OPEN OF C_IXCOL FAILED, SQL-CODE: -xxx

Explanation: An error was detected during opening a cursor for SYSIBM.SYSKEYS joined with SYSIBM.SYSINDEXES and SYSIBM.SYSTABLES. If the SQL code is -911 or -913, Easy Explain waits a short period and then tries to reprocess the current statement in the plan. Refer to message ANL0532E. For other SQL codes, Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

User response: Reissue the request when the resource is available.

Severity: 8
in the plan. Refer to message ANL0532E. For other SQL
codes, Easy Explain continues with the current request
after printing the formatted SQLCA and the error
message.

User response: Reissue the request when the resource
is available.

Severity: 8

ANL0526E  FETCH FROM C_IXCOL FAILED,
           SQL-CODE: -xxx

Explanation: An error was detected during fetch from
SYSIBM.SYSKEYS joined with SYSIBM.SYSINDEXES and
SYSIBM.SYSTABLES. If the SQL code is -911 or
-913, Easy Explain waits a short period and then tries
to reprocess the current statement in the plan. Refer to
message ANL0532E. For other SQL codes, Easy Explain
continues with the current request after printing the
formatted SQLCA and the error message.

User response: Reissue the request when the resource
is available.

Severity: 8

ANL0528E  OPEN OF C_STMT FAILED,
           SQL-CODE: -xxx

Explanation: An error was detected during opening a
cursor for SYSIBM.SYSTMT. A formatted print of the
SQLCA and an error message is provided. Easy Explain stops
processing the current request and continues with the
next input record.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0530E  FETCH FROM C_STMT FAILED,
           SQL-CODE: -xxx

Explanation: An error was detected during a fetch from
SYSIBM.SYSTMT. A formatted print of the SQLCA and an error message is provided. Easy Explain continues processing the current request.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0532E  A DEADLOCK OR TIMEOUT HAS
          OCCURRED - THE STATEMENT IS
          REPROCESSED

Explanation: An SQL code of -911 or -913 was
encountered. Easy Explain stops processing the current
miniplan or statement. A ROLLBACK call is issued to
close any open cursors. After a short delay, Easy

Explain reprocesses the failing statement in the current
plan.

User response: Reissue the request when the resource
is available.

Severity: 8

ANL0534E  A DEADLOCK OR TIMEOUT HAS
          OCCURRED 3 TIMES WITHIN THIS
          PLAN

Explanation: An SQL code of -911 or -913 was
encountered three times within processing of the same
plan. A formatted print of the SQLCA and the error
message is provided. Easy Explain stops processing the
current plan, and continues with the next input record.

User response: Reissue the request when the resource
is available.

Severity: 8

ANL0536E  EXPLAIN OF A REMOTE OBJECT IS
          NOT ALLOWED, SQL-CODE: -xxx

Explanation: An immediate EXPLAIN statement failed
because the statement being explained contained a
table located in another DB2 system. The EXPLAIN
statement can only contain local objects. Easy Explain
continues with the next input record.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0538E  EXPLAIN OF SUPPLIED
          SQL-STATEMENT FAILED, SQL-CODE:
          -xxx

Explanation: An immediate EXPLAIN statement failed.
A formatted print of the SQLCA and an error message is provided together with the statement to be explained. Easy Explain continues with the next input record.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0540E  TABLE SPACE dbname.tsname
          NOT FOUND IN SYSIBM.SYSTABLESPACE -
          SQL-CODE: -xxx

Explanation: Easy Explain was not able to find the
named table space in SYSIBM.SYSTABLESPACE.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.
ANL0544E  COLUMN colname NOT FETCHED FROM SYSIBM.SYSCOLUMNS - SQL-CODE: -xxx

Explanation: Easy Explain was not able to find the named table in SYSIBM.SYSCOLUMNS. If the SQL code is -911 or -913, Easy Explain waits a short period and then tries to reprocess the current statement in the plan. Refer to message ANL0532E. For other SQL codes, Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0546E  INDEX authid.ixname NOT FOUND IN SYSIBM.SYSINDEXES - SQL-CODE: -xxx

Explanation: Easy Explain was not able to find the named index in SYSIBM.SYSINDEXES. A formatted print of the SQLCA and an error message is provided.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0548E  PLAN planname NOT FOUND IN SYSIBM.SYSPLAN - SQL-CODE: -xxx

Explanation: Easy Explain was not able to find the named plan in SYSIBM.SYSPLAN.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0550E  OPEN OF C_KEYDIST FAILED, SQL-CODE: -xxx

Explanation: An error was detected during opening a cursor for SYSIBM.SYSFIELDS. If the SQL code is -911 or -913, Easy Explain waits for a short time and tries to reprocess the current statement in the plan. See message ANL0532E. For other SQL codes, Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0552E  OPEN OF C_COLDIST FAILED, SQL-CODE: -xxx

Explanation: An error was detected during opening a cursor for SYSIBM.SYSCOLDIST. If the SQL code is -911 or -913, Easy Explain waits a short period and then tries to reprocess the current statement in the plan. Refer to message ANL0532E. For other SQL codes, Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0554E  DATABASE dbname NOT FOUND IN SYSIBM.SYSDATABASE - SQL-CODE: -xxx

Explanation: Easy Explain was not able to find the named database in SYSIBM.SYSDATABASE.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0560E  OPEN OF C_TSKEYS FAILED, SQL-CODE: -xxx

Explanation: An error was detected during opening a cursor for SYSIBM.SYSKEYS. If the SQL code is -911 or -913, Easy Explain waits a short period and then tries to reprocess the current statement in the plan. Refer to message ANL0532E. For other SQL codes, Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0570E  CURRENT APPLICATION ENCODING SCHEME COULD NOT BE FOUND - WILL USE 037 AS DEFAULT

Explanation: An error was detected when Easy Explain tried to select the current application encoding scheme. Easy Explain continues operation by using 037 as CCSID. This message is only issued for DB2 Version 8 and later releases.

User response: Specify a valid CCSID, and issue the request again.

Severity: 8
ANL0572E  CONVERSION FROM UNICODE TO EBCDIC FAILED - RC = xx, REASON = yyy
Explanation: An error was detected during conversion of Unicode to EBCDIC. Easy Explain continues operation. This message is only issued for DB2 Version 8 and later releases.
User response: Refer to the explanation for the RC and REASON codes in z/OS Support for Unicode: Using Conversion Services.
Severity: 8

ANL0580E  UNAUTHORIZED ACCESS TO THE Q.OBJECT_DIRECTORY TABLE - REQUEST IGNORED
Explanation: You have specified an input record with the QMF= keyword. DB2 has found that the Easy Explain plan does not have select authorization to the Q.OBJECT_DIRECTORY table and returns an SQL code of -551 or -552. Easy Explain proceeds to the next input record.
User response: Obtain authorization to access the Q.OBJECT_DIRECTORY table, and issue the request again.
Severity: 8

ANL0582E  UNAUTHORIZED ACCESS TO THE Q.OBJECT_DATA TABLE - REQUEST IGNORED
Explanation: You have specified an input record with the QMF= keyword. DB2 has found that the Easy Explain plan does not have select authorization to the Q.OBJECT_DATA table and returns an SQL code of -551 or -552. Easy Explain proceeds to the next input record.
User response: Obtain authorization to access the Q.OBJECT_DATA table, and issue the request again.
Severity: 8

ANL0584E  ACCESS TO TABLE Q.OBJECT_DIRECTORY FAILED - SQL-CODE: -xxx
Explanation: You have specified an input record with the QMF= keyword. DB2 has found that the Easy Explain plan does not have proper authorization to the Q.OBJECT_DIRECTORY table. A formatted print of the SQLCA and an error message is provided.
User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual. Easy Explain proceeds to the next input record.
Severity: 8

ANL0586E  ACCESS TO TABLE Q.OBJECT_DATA FAILED - SQL-CODE: -xxx
Explanation: You have specified an input record with the QMF= keyword. DB2 has found that the Easy Explain plan does not have proper authorization to the Q.OBJECT_DATA table. A formatted print of the SQLCA and an error message is provided.
User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual. Easy Explain proceeds to the next input record.
Severity: 8

ANL0590E  THE LOCATION FOR THE PLAN_TABLE TABLE COULD NOT BE OBTAINED - SQL-CODE: -xxx
Explanation: The installation has specified a location (database and table space name) for all newly created PLAN_TABLE tables. Easy Explain is not able to select this location information from the table EEWORK (or view EEE_LOCATION). The processing continues, but the PLAN_TABLE table is created in the default database.
User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual. Easy Explain proceeds to the next input record.
Severity: 8

ANL0592E  EITHER dbname OR tsname FOR THE PLAN_TABLE TABLE IS INVALID
Explanation: You have specified an invalid location (database and table space name) for all newly created PLAN_TABLE tables. Easy Explain is not able to understand the database name or the table space name (one of the names might be blanks). The processing continues, but the PLAN_TABLE table is created in the default database.
User response: Specify a valid location (dbname or tsname) for the PLAN_TABLE tables, and issue the request again.
Severity: 8

ANL0600E  THE EXPLAIN TABLE authid.PLAN_TABLE WAS NOT CREATED - SQL-CODE: -xxx
Explanation: Easy Explain tried to create a PLAN_TABLE for user authid, but failed. The reason is probably that the CREATETAB authorization is missing. A formatted print of the SQLCA and an error message is provided. Easy Explain stops processing. This message is preceded by message ANL0100E.
User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 12

ANL0602E ADDITION OF THREE V3R1 COLUMNS TO authid.PLAN_TABLE FAILED - SQL-CODE: -xxx

Explanation: Easy Explain tried to ALTER the three columns PREFETCH, COLUMN_FN_EVAL, and MIXOPSEQ to the existing PLAN_TABLE. One or more of the three ALTER statements failed. Easy Explain continues processing using the pre-V3R1 format of the PLAN_TABLE.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0618E PLAN_TABLE IS NOT COMPATIBLE WITH DB2 REL. x.y - SQL-CODE: -xxx

Explanation: The format of the PLAN_TABLE is not compatible with the current release of DB2.

User response: Issue an SQL=-statement. This updates the PLAN_TABLE to the same level as your DB2 system. Then reissue the explain request.
Severity: 8

**ANL0620E** UPDATE OF DYNAMICALLY EXPLAINED DATA FAILED,
SQL-CODE: -xxx

**Explanation:** Easy Explain has detected that the plan or package being explained was bound with the EXPLAIN(NO) option. Easy Explain is consequently issuing a dynamic Explain of the individual statements. After this dynamic explain, Easy Explain is updating the produced rows in the PLAN_TABLE table, but the update failed. Easy Explain continues processing the current request.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

**Severity:** 8

**ANL0630E** OPEN OF C_AUTH FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for binder.EEE_EEAUTH. Default authorization values are used (see message ANL0160I). Easy Explain continues processing the current request.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

**Severity:** 8

**ANL0636E** OPEN OF C_EEPL FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for binder.EEE_ANLPLAN. Printing of plan history information is stopped. Easy Explain continues normal processing.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

**Severity:** 8

**ANL0638E** OPEN OF C_EEED1 FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for binder.EEE_EEEDBRM. Printing of DBRM information is stopped. Easy Explain continues normal processing.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

**Severity:** 8

**ANL0640E** SELECT FROM EEEPLAN FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected when selecting a row from binder.EEE_ANLPLAN. Saving plan information is stopped. Easy Explain continues normal processing.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

**Severity:** 8

**ANL0642E** OPEN OF C_EEED2 FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for binder.EEE_EEEDBRM. Saving of DBRM information is stopped. Easy Explain continues normal processing.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

**Severity:** 8

**ANL0644E** FETCH FROM C_EEED1 FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for binder.EEE_EEEDBRM. Saving of DBRM information is stopped. Easy Explain continues normal processing.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

**Severity:** 8

**ANL0646E** FETCH FROM C_EEED2 FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for binder.EEE_EEEDBRM. Saving of DBRM information is stopped. Easy Explain continues normal processing.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

**Severity:** 8

**ANL0650E** MAXIMUM HINT ID COULD NOT BE FOUND - SQL-CODE: -xxx

**Explanation:** Easy Explain was not able to find the maximum hint ID number used in the EEEHINT table. Consequently, Easy Explain was not able to assign a hint ID number for the current operation. No hint ID was assigned. Easy Explain continues normal processing.
User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0652E  INSERT IN EEEHINT TABLE FAILED - SQL-CODE: -xxx

Explanation: Easy Explain was trying to store the current hint ID number in the EEEHINT table, but the operation did not succeed. Easy Explain continues normal processing.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL064E  ADDITION OF TWO V7R1 COLUMNS TO authid PLAN_TABLE FAILED - SQL-CODE: -xxx

Explanation: Easy Explain tried to ALTER the two columns PARENT_QBLOCKNO and TABLE_TYPE to the existing PLAN_TABLE. One or more of the two ALTER statements failed. Easy Explain continues processing using the pre-V7 format of the PLAN_TABLE.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL066E  ADDITION OF SEVEN V8R1 COLUMNS TO authid PLAN_TABLE FAILED - SQL-CODE: -xxx

Explanation: Easy Explain tried to ALTER the seven columns TABLE_ENCODE, TABLE_SCCSID, TABLE_MCCSID, TABLE_DCCSID, ROUTINE_ID, CTEREF, and STMTTOKEN to the existing PLAN_TABLE. One or more of the seven ALTER statements failed. Easy Explain continues processing using the pre-V8 format of the PLAN_TABLE.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0660E  ADDITION OF THREE V5R1 COLUMNS TO authid PLAN_TABLE FAILED - SQL-CODE: -xxx

Explanation: Easy Explain tried to ALTER the three columns WHEN_OPTIMIZE, QBLOCK_TYPE, and BIND_TIME to the existing PLAN_TABLE. One or more of the three ALTER statements failed. Easy Explain continues processing using the pre-V5 format of the PLAN_TABLE.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0662E  ADDITION OF THREE V6R1 COLUMNS TO authid PLAN_TABLE FAILED - SQL-CODE: -xxx

Explanation: Easy Explain tried to ALTER the three columns OPTHINT, HINT_USED, and PRIMARY_ACCESSTYPE to the existing PLAN_TABLE. One or more of the three ALTER statements failed. Easy Explain continues processing using the pre-V6 format of the PLAN_TABLE.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0680E  ALTER TABLE ... COLUMN FAILED FOR authid PLAN_TABLE - SQL-CODE: -xxx

Explanation: Easy Explain tried to ALTER ten columns of the existing PLAN_TABLE. One or more of the column alters failed. Easy Explain continues processing using the pre-V8 format of the PLAN_TABLE.

User response: Check the non-zero SQL code (-xxx), and follow the instructions given in the DB2 Messages and Codes manual.

Severity: 8

ANL0682E  ALTER TABLE ... COLUMN FAILED FOR authid DSN_STATEMNT_TABLE - SQL-CODE: -xxx

Explanation: Easy Explain tried to ALTER two columns of the existing DSN_STATEMNT_TABLE. One or more of the column alters failed. Easy Explain
continues processing using the pre-V8 format of the DSN_STATEMNT_TABLE.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the *DB2 Messages and Codes* manual.

**Severity:** 8

---

**ANL0684E** ALTER TABLE ... COLUMN FAILED
FOR authid.DSN_FUNCTION_TABLE - SQL-CODE: -xxx

**Explanation:** Easy Explain tried to ALTER nine columns of the existing DSN_FUNCTION_TABLE. One or more of the column alters failed. Easy Explain continues processing using the pre-V8 format of the DSN_FUNCTION_TABLE.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the *DB2 Messages and Codes* manual.

**Severity:** 8

---

**ANL0690E** ADDITIONAL xxx STATEMENTS
WITH NUMBERS GREATER THAN 32767 WERE FOUND

**Explanation:** The plan or package explained had SQL statements with statement numbers greater than 32767. These statements are all given a statement number of 0 (zero) in the DB2 SYSSTMT / SYSPACKSTMT catalog tables and can therefore not be related to the individual corresponding rows in the PLAN_TABLE table. The given number of statements are not shown in the Easy Explain report.

**User response:** Split the application code into smaller source code modules before precompiling takes place.

**Severity:** 8

---

**ANL0704E** SELECT FROM SYSIBM.SYSPACKLIST
FAILED - SQL-CODE: -xxx

**Explanation:** An error was detected during selecting from SYSIBM.SYSPACKLIST. Easy Explain continues with the next input statement, after printing the formatted SQLCA and the error message.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the *DB2 Messages and Codes* manual.

**Severity:** 8

---

**ANL0708E** OPEN OF C_PACKAGE_NAVN FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for SYSIBM.SYSPACKAGE. Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the *DB2 Messages and Codes* manual.

**Severity:** 8

---

**ANL0710E** OPEN OF C_PACK_STMT FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for SYSIBM.SYSPACKSTMT. Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the *DB2 Messages and Codes* manual.

**Severity:** 8

---

**ANL0712E** FETCH FROM C_PACK_STMT FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during fetching a row from SYSIBM.SYSPACKSTMT. Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the *DB2 Messages and Codes* manual.

**Severity:** 8

---

**ANL0714E** OPEN OF C_SYSPACK_STMT FAILED,
SQL-CODE: -xxx

**Explanation:** An error was detected during opening a cursor for SYSIBM.SYSPACKSTMT. Easy Explain continues with the current request after printing the formatted SQLCA and the error message.

**User response:** Check the non-zero SQL code (-xxx), and follow the instructions given in the *DB2 Messages and Codes* manual.

**Severity:** 8
ANL0716E  OPEN OF C_PACK1 FAILED,
SQL-CODE: -xxx

Explanation: An error was detected during opening a
cursor for SYSIBM.SYSPACKLIST. Easy Explain
continues with the current request after printing the
formatted SQLCA and the error message.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0718E  OPEN OF C_PACK2 FAILED,
SQL-CODE: -xxx

Explanation: An error was detected during opening a
cursor for SYSIBM.SYSPACKAGE. Easy Explain
continues with the current request after printing the
formatted SQLCA and the error message.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0720E  CURRENT SERVER NAME COULD
NOT BE ACQUIRED - SQL-CODE: -xxx

Explanation: An error was detected during selecting
the CURRENT SERVER special register. Easy Explain
continues processing using 16 blanks as the current
server identification.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0722E  CONNECT RESET FAILED -
SQL-CODE: -xxx

Explanation: An error was detected during a
CONNECT RESET operation. Easy Explain continues
processing using the current specified server.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0724E  CONNECT TO xxxx FAILED -
(NON-CONNECTABLE STATE) -
SQL-CODE: -752

Explanation: An error was detected during a
CONNECT TO :location operation. Easy Explain issues
a Connect Reset and continues processing on the home
server location.

User response: Specify a valid location, and issue the
request again.

Severity: 8

ANL0726E  CONNECT TO xxxx FAILED -
(SERVER NAME NOT KNOWN) -
SQL-CODE: -950

Explanation: An error was detected during a
CONNECT TO :location operation. Easy Explain issues
a Connect Reset and continues processing on the home
server location.

User response: Specify a valid location, and issue the
request again.

Severity: 8

ANL0728E  CONNECT TO xxxx FAILED -
SQL-CODE: -xxx

Explanation: An error was detected during a
CONNECT TO :location operation. The specified server
name is acceptable and the process is in a connectable
state. Easy Explain issues a Connect Reset and
continues processing on the home server location.

User response: Check the non-zero SQL code (-xxx),
and follow the instructions given in the DB2 Messages
and Codes manual.

Severity: 8

ANL0730E  SERVER IS NOT A DB2 SUBSYSTEM -
WILL ISSUE A CONNECT RESET

Explanation: A connect to a remote server location has
been established, but Easy Explain has found that the
remote server is not a DB2 system. Easy Explain issues
a CONNECT RESET command to connect to the home
server location.

This message is preceded by message ANL0180I that
explains the server type. Easy Explain continues
processing at the home location.

User response: Verify that the database on the remote
server is a DB2 system. If not, specify a valid location,
and issue the request again.

Severity: 8

ANL080E  The value must not contain embedded
blanks.

Explanation: The column name contains blanks.

User response: Specify a column name without
blanks.
**ANL081E** User `user_id` needs the privilege for
SELECT from SYSIBM.SYSINDEXES
and SYSIBM.SYSCOLUMNS, CREATE
INDEX, and CREATE TABLE
owner_name.DSN_VIRTUAL_INDEXES,
if DSN_VIRTUAL_INDEXES table is
not created at SQL PA customization.

**Explanation:** If the DSN_VIRTUAL_INDEXES table was not created when DB2 SQL Performance Analyzer was customized, the user needs SELECT privilege to access or create tables.

**User response:** Ensure that the user has the required privilege.

**ANL082E** NO cannot be specified for the
INCLUDE NULL KEYS parameter when
the `key_column` key column does not allow null values.

**User response:** Specify YES for the INCLUDE NULL KEYS parameter.

**ANL083E** NO cannot be specified for the
INCLUDE NULL KEYS parameter when
YES is specified for the PARTITIONED parameter.

**User response:** Specify YES for the INCLUDE NULL KEYS parameter.

**ANL084E** NO cannot be specified for the
INCLUDE NULL KEYS parameter when
YES is specified for the UNIQUE parameter.

**User response:** Specify YES for the INCLUDE NULL KEYS parameter.

**ANL086W** The user-level user parameter data set is unavailable or does not exist. The job cards cannot be saved.

**Explanation:** The job cards were not saved because the specified data set is not available or has not been created.

**User response:** Ensure that the user-level user parameter file exists and is available for updates.

**ANL087I** The job cards were saved.

**User response:** No action is required.

**ANL088I** The user-level user parameter data set must be created before the job cards can be saved. To create the data set with the displayed name and settings, press Enter. Otherwise, change the name, settings, or both, and press Enter.

**ANL089I** The user-level user parameter data set was created. Press Enter to save the job cards, or press PF3 to cancel the changes.

**Explanation:** The data set was created, and the job cards can be saved now.

**User response:** To save the job cards, press Enter. To cancel the changes, press PF3.

**ANL0880E** DB2 release level unknown - release level assumed to be V8

**Explanation:** Easy Explain was not able to determine the version and release level of the current DB2 subsystem. Easy Explain continues processing assuming that the version and release level is 7.1.

**User response:** None.

**Severity:** 8

**ANL0900E** EASY EXPLAIN ABNORMALLY TERMINATES WITH RETURN CODE: 16

**Explanation:** The Easy Explain On Error Unit has been activated. A snap dump is being produced. Examine the dump to determine the reason for the error. If the Easy Explain program has failed, a PL/I On Code is printed in the snap dump. Easy Explain stops processing.

**User response:** Contact the IBM Software Support Center.

**Severity:** 16

**ANL1000W** PARSING IS INCOMPLETE AND IN DOUBT FOR STATEMENT:
`nnnnnnnnn`.PARSING FAILED, BUT PROCESSING CONTINUES.
ERRCODE = ccccc.

**Explanation:** The parser has encountered a sequence of SQL clauses it is unable to handle, possibly due to an incorrect SQL statement combining clauses that are illegal or the parser did not anticipate their use together. The parser gives up and the processing continues, although the cost estimate for this statement might be inaccurate.

**User response:** Check SQL with EXPLAIN to see if it is valid. If valid, contact IBM Software Support.
ANL1001E  A processing error was encountered during SQL statement analysis. ANLOUT output trace file was not defined.

Explanation: One of the files required for processing was not properly allocated. ANLOUT holds the SQL used for reporting and display after the parser checks syntax.

User response: Check for an ANLOUT DD card in batch JCL. For all other environments, contact IBM Software Support.

ANL1002E  ALIGNMENT ERROR IN DBRM READ, RELEASE = c, DID NOT POSITION ON RECORD. The record rec is bypassed and the next valid record in the DBRM is searched for.

Explanation: During the processing of DBRM input, the parser has incorrectly read the offsets of blocks within, such that it is no longer properly positioned on a valid control block inside the DBRM. SQL PA supports several versions of DBRM and the recognition is automatic (RELEASE = c indicates the version).

User response: Verify that the DBRM is stable and error free. Then try running it again.

ANL1003E  End of DBRM file detected before end of records indicated there. Pgm terminates because the DBRM module is unusable. Check for completeness and rebind if necessary, then try again.

Explanation: A DBRM module indicated several blocks to follow, but the EOF marker was found before the last block indicated might need to rebind DBRM.

User response: Verify that the DBRM is stable and error free. Then try running it again.

ANL1004E  A processing error was encountered during SQL statement analysis. ANLIN input SQL file has not been defined.

Explanation: The input file required for processing was not properly allocated or defined. ANLIN holds the SQL used for analysis and is a sequential or PDS member, or an entire PDS when DBRMKEY parameter is also active.

User response: Check for an ANLIN DD card in batch JCL. For TSO, check Input Data Set Name on the Main panel to make sure it exists. It cannot be an empty member. For all other environments, contact IBM Software Support.

ANL1007I  UNABLE TO SET CURRENT SQUID TO SQL PA SECONDARY AUTHID: xxxxxxxx. PGM WILL USE PLAN_TABLE UNDER PRIMARY AUTHID yyyyyyyyy INSTEAD. AUTOMATIC CORRECTIVE ACTION TAKEN BY PGM. Processing continues.

Explanation: An SQL error code of -553 was encountered, indicating the unavailability of secondary authorization IDs, ANLUSER1 - ANLUSERn. SQL PA automatically reverts to the primary authorization ID's plan table, if available.

User response: Check the assignment of secondary authorization IDs in the ANLUSER0.REGISTRY table. Ensure that the primary authorization ID owns a PLAN_TABLE.

ANL1008E  THERE ARE NO AVAILABLE ANL SECONDARY AUTHIDS UNUSED AT THIS TIME. NO PLAN_TABLE IS AVAILABLE BECAUSE EITHER: (A) ALL ENTRIES IN REGISTRY ARE IN USE (B) NO REGISTRY WAS CREATED ON THIS SUBSYSTEM (C) NO GENERIC PLAN_TABLES EXIST, AND USER DOES NOT OWN ONE HINT: TRY THE USEPLAN PARAMETER TO SPECIFY A PLAN_TABLE OR NOTIFY SYSTEMS TO EXPAND THE NUMBER OF ANL IDS FOR SIMULTANEOUS USAGE. PGM TERMINATES BECAUSE THE REGISTRY OR THE PLAN_TABLE IS UNAVAILABLE. TRY AGAIN LATER.

Explanation: An SQL error code of 100 was encountered during registration of the ANLUSERn secondary authorization ID—all PLAN_TABLEs were busy during the time SQL PA tried to start. Either there is an insufficient number of ANLUSERn authorization IDs available for the number of simultaneous users (usually 10), or an error has left all the ANLUSERn authorization IDs hung as in use erroneously. Normally, SQL PA automatically resets these entries when full.

User response: Assign more secondary authorization IDs if fewer than 10; otherwise run ANLRESET to free up hung entries in the ANLUSER0.REGISTRY table. If the problem persists, contact IBM Software Support.
ANL1009W  PGM CANNOT DETERMINE THE
COST OF STATEMENT: nnnnnnnn,
SQLCODE = cccc. ONE OR MORE
HOST VARIABLES CANNOT BE
CORRECTLY SUBSTITUTED WITH
MARKER. PARAMETER MARKERS (?)
ARE ILLEGAL IN SOME EXPLAIN
REQUESTS. Or, AN ILLEGAL SYMBOL
HAS BEEN ENCOUNTERED WITHIN
THE SQL. The statement is ignored and
processing continues.

Explanation: One of several SQL error codes were encountered: -104, -312 or -314, all indicating an incompatibility with the parameter marker and statement processed. Explain does not allow ? to hold place on both sides of expression, as part of concatenation, and several other instances. SQL PA catches use of ? in the SELECT list, and that is corrected dynamically. The other uses of ? are not allowed. SQL PA does not cost this query, but continues to process others.

User response: Recode the SQL to use allowed predicate structure. Or replace the ? with your guess of the proper values and run again.

ANL1010W  PGM CANNOT DETERMINE THE
COST OF STATEMENT: nnnnnnnn,
SQLCODE = cccc. THE PREPARED
FORM OF THE STATEMENT WAS
UNACCEPTABLE TO DB2. The
statement is ignored and processing
continues.

Explanation: An SQL error code -518 or -084 was encountered, indicating DB2 could not prepare the statement because of its SQL content. SQL PA moves on to the next explainable statement. Sometimes, a -518 is caused by SQL not properly aligned in columns 1-72, in which case SQL PA scans only part of the query.

User response: Check the input file and correct any errors.

ANL1011E  SQL PA cannot find an explanation
table for user_ID because the table is not
in the correct format. The SQL code is
SQL_CODE. Processing has stopped.

Explanation: An SQL error code of 219 or -219 was encountered, indicating that SQL PA was not properly initialized, and one or more required tables cannot be accessed.

User response: Correct the explanation table and run again.

ANL1012E  SQL PA cannot use an explanation table
for user_ID because the table is not in
the correct format. The SQL code is
SQL_CODE. Processing has stopped.

Explanation: The explanation table was not correctly created or is not at the proper level to support this release of DB2. SQL error codes -221, -200, or 220 can occur.

User response: Correct the explanation table and run again.

ANL1013E  PGM TRYING TO USE
authid.PLAN_TABLE, SQLCODE = cccc.
AN AUTHORIZATION ERROR WAS
ENCOUNTERED. PGM DOES NOT
HAVE PROPER ACCESS. PGM
TERMINATES THIS RUN.

Explanation: An SQL error code -922 was encountered, indicating that SQL PA was not properly installed, and one or more required tables cannot be accessed.

User response: Authorize SQL PA to access the tables. If the error continues, retrace SQL PA steps.

ANL1014W  PGM CANNOT DETERMINE THE
COST OF STATEMENT: nnnnnnnn,
SQLCODE = cccc. THE PREPARED
STATEMENT IS TOO LONG OR
COMPLEX FOR DB2 TO HANDLE. The
statement is ignored and processing
continues.

Explanation: SQL PA got an SQL error code -101 indicating that DB2 cannot parse the SQL statement, which is too long or too complex. SQL PA continues with other SQL statements in the run.

User response: Correct and run the SQL statement again.

ANL1015W  PGM CANNOT DETERMINE THE
COST OF STATEMENT: nnnnnnnn,
SQLCODE = cccc. THE PREPARED
STATEMENT REFERS TO A REMOTE
OBJECT NOT ON THIS SYSTEM. The
statement is ignored and processing
continues.

Explanation: SQL PA does not do a remote Explain. Therefore, it cannot explain distributed SQL statements that require objects on other systems and catalogs. SQL PA continues to the next statement.

User response: Run again, using VIADRDA to route the SQL to the remote system for processing.
ANL1016E  A fatal SQL error was encountered during product registration. SQLCODE = ccccc, IOS OPERATION = x.

Explanation: SQL PA was unable to register and assign an ANLUSER n authid for SQL PA processing. The REGISTRY table might be damaged or inoperable. See additional reason codes.

User response: Start ANLRESET on the ANLUSER0.REGISTRY table, and run again.

The message might also show:
INCOMPATIBILITY BETWEEN DBRM & PROGRAM IS INDICATED:
REBIND ANLSQLPA... only if SQLCODE -818
ERROR OCCURRED DURING DECLARE FOR
REGISTRY CURSOR... if IOS = 1
ERROR OCCURRED DURING SET CURRID TO
CURRENT SQILD... if IOS = 2
ERROR OCCURRED DURING OPENING OF
REGISTRY CURSOR... if IOS = 3
ERROR OCCURRED DURING FETCH OF
REGISTRY ENTRIES... if IOS = 4
ERROR OCCURRED DURING UPDATING OF
REGISTRY TABLE... if IOS =5
ERROR OCCURRED DURING COMMIT OF
REGISTRY UPDATE... if IOS = 6
ERROR OCCURRED DURING SET CURRENT
SQILD TO ANLID... if IOS = 7
ERROR OCCURRED USING THE PRIMARY
AUTHID PLAN_TABLE... if IOS = 8
ERROR OCCURRED DURING CAF CONNECTION
(OPEN OR CLOSE)... if IOS = 9
ERROR OCCURRED DURING RESET REGISTRY
ENTRIES > 1 HR OLD... if IOS = 10
ERROR OCCURRED DURING READING OF
ANLCNTL PARAMETERS... if IOS = 11
ERROR OCCURRED DURING READING OF
ANLPARM PARAMETERS... if IOS = 12
ERROR OCCURRED DURING USE OF SETPLAN
AUTHID PLAN_TABLE... if IOS = 20
ERROR OCCURRED DURING ASSIGNMENT OF
SETPLAN AUTHID P_T... if IOS = 21
ERROR OCCURRED BECAUSE SETPLAN
PLAN_TABLE IS NOT USABLE... if IOS = 22

ANL1017W  THE INPUT FILE SPECIFIED BY ANLIN IS TOO LONG... SOME TRUNCATION OR FILLER MAY OCCUR DUE TO RECORD SIZE. Processing continues. COND = ccccc.

Explanation: The input to SQL PA was not a DBRM module or a flat, sequential file with LRECL = 80, RECFM = FB. While possible to handle other record sizes, typically 80 is specified for operational use.

User response: None; use 80 character input in future runs.

ANL1019W  PGM CANNOT DETERMINE THE COST OF STATEMENT: xxxxxxxx, SQLCODE = uuuu LACK AUTHORITY TO PERFORM PREPARE/EXPLAIN ON OBJECTS IN QUERY. The statement is ignored and processing continues.

Explanation: The person starting SQL PA does not have prepare or explain privileges on one or more objects in query. SQLCODE returns -551.

User response: The running authorization ID does not have authority to issue an EXPLAIN on table. Use the SQL statement GRANT SELECT ON TABLE tablenameto authid to allow the user to access the table.

ANL1020E  PROCESSING ERROR ENCOUNTERED DURING SQL STATEMENT ANALYSIS. NO EXPLAINABLE SQL STATEMENTS WERE FOUND.

Explanation: The input SQL did not contain a valid SELECT, INSERT, UPDATE, DELETE, or REFRESH statement. Parser syntax failed to find an explainable SQL statement suitable for analysis.

User response: Check SQL for an explainable statement.

ANL1021E  PROCESSING ERROR WAS ENCOUNTERED DURING SQL STATEMENT ANALYSIS. MORE THAN 4096 RECORDS FOR THE APAS ARRAY.

Explanation: One of the internal array structures used to hold parser intelligence has overflowed.

User response: Save output and contact IBM Software Support. Must be corrected programmatically.

ANL1022E  PROCESSING ERROR WAS ENCOUNTERED DURING SQL STATEMENT ANALYSIS. MORE THAN maximum_number RECORDS FOR THE ARELS ARRAY.

Explanation: One of the internal array structures used to hold referential integrity information has overflowed. For ANLSQLPA, the maximum number of records is 4096. For ANLQMF and the stored procedures, the maximum number of records is 512.

User response: Save output and contact IBM Software Support.

ANL1023E  PROCESSING ERROR WAS ENCOUNTERED DURING SQL STATEMENT ANALYSIS. MORE THAN maximum_number RECORDS FOR THE AEXP ARRAY.
Explanation: One of the internal array structures used to hold access plan data has overflowed. For ANLSQLPA, the maximum number of records is 4096. For ANLQMF and the stored procedures, the maximum number of records is 512.

User response: Save output and contact IBM Software Support.

ANL1024E  PROCESSING ERROR WAS ENCOUNTERED DURING SQL STATEMENT ANALYSIS. MORE THAN maximum_number RECORDS FOR THE ASTAT ARRAY.

Explanation: One of the internal array structures used to hold catalog statistics has overflowed. For ANLSQLPA, the maximum number of records is 4096. For ANLQMF and the stored procedures, the maximum number of records is 512.

User response: Save output and contact IBM Software Support.

ANL1025I Because the qualify parameter was set to blanks, PGM assumes synonym use. PGM will use the PLAN_TABLE under the primary AUTHID aaaaaaaaa for this run. Automatic synonyms reaction is taken by PGM. Processing continues.

Explanation: A blank QUALIFY parameter was specified indicating an intent to use synonym names in the SQL. SQL PA cannot switch to a generic PLAN_TABLE in this case because the statement must be prepared under the creator of the Synonyms (the only valid user of same), a DB2 requirement. This message is only informational, but puts user on notice that a valid plan table must exist under the running authorization ID to proceed.

User response: Verify that a valid plan table exists, and run again.

ANL1026W SQL PA CANNOT DETERMINE THE COST OF STATEMENT: nnnnnnnn, SQLCODE = cccc. THE TABLE NAME IS UNDEFINED OR ELSE REFERENCES A SYNONYM NOT DETECTED. STATEMENT IGNORED AND PROCESS CONTINUES.

Explanation: An SQL error code -204 was encountered, where DB2 was unable to PREPARE the SQL statement, due to its inability to recognize the table name or high-level qualifier; this is a DB2 limitation in certain prepares. SQL PA ignores the statement and continues. This message occurs in QMF version only.

User response: Fix the SQL statement and run again.

ANL1027I UNNECESSARY USE OF SEMICOLON TO END QMF STATEMENT. CHARACTER IGNORED (SQLCODE +98) AND SQL PA CONTINUES.

Explanation: A QMF statement ended with a semicolon (;), which is not necessary under QMF. The Prepare routine issued the +98 error code, and SQL PA removed the semicolon and prepared the statement again automatically.

User response: None.

One of the following unnumbered messages could occur in QMF:

- **ONE OR MORE LIMITS EXCEEDED BY YOUR QUERY PER SQL PA GOVERNOR EXIT or ONE OR MORE LIMITS EXCEEDED BY YOUR QUERY–ANL HAS TERM I/O ERROR.**

  Your query exceeded the installation coded limits for CPU, Elapsed Time or I/O estimates. Written to the QMF log. You might be given the option to cancel this query before starting. Term I/O error if DB2 cannot ask you WOULD YOU LIKE SQL PA TO CANCEL THIS QUERY (Y/N)?

- **YOUR QUERY HAS BEEN CANCELLED BY THE SQL PA GOVERNOR EXIT.**

  The SQL PA Governor can prevent queries from running. The installation might choose to enforce this option when installing ANLGOV1 or leave the decision to the user. In either case, run the query in Batch mode.

ANL1028W ERROR ACCESSING ANLNl.DSN_STATEMNT_TABLE, SQLCODE = ssssss. PGM Ignores TABLE AND CONTINUES PROCESSING QUERIES. QUERYNO NOT FOUND WAS nnnnnn, PlanCount = ccccc.

Explanation: SQL PA could not access the optional generic DSN_STATEMNT_TABLE, because it was not created or was damaged (possible SQL return codes are -312, -204 or +100). Users who specify PRECISE YES want to use the Optimizer's Instruction Count (Path Length) estimate, provided by this Explain table. SQL PA ignores the table and moves on, using its own estimate instead. In Version 8 systems, SQL PA attempts to create one before abandoning efforts.

User response: If needed, create DSN_STATEMNT_TABLE and run the query again.

ANL1029E A processing error was encountered during SQL statement analysis. The SYSPRINT output log file was not defined.

Explanation: One of the files required for processing was not properly allocated. SYSPRINT holds the output
messages and Cost Summary analysis for all SQL analyzed.

User response: Check for a SYSPRINT DD card in batch JCL. For all other environments, contact IBM Software Support.

---

**ANL1030E**  ANLCNTL FILE NOT ALLOCATED.

**Explanation:** This message is associated with the SQL PA stored procedure. An ANLCNTL DD statement is not defined in the stored procedure address space (DSNSPAS or WLMSPAS) startup JCL. SQL PA is unable to use any installation defined parameters to describe the configuration.

User response: Add an ANLCNTL DD statement identifying the target host configuration file to the DSNSPAS or WLMSPAS procedure in SYS1.PROCLIB.

---

**ANL1031W**  DB2 OBJECT NOT FOUND (TABLE OR COLUMN NAME) queryno, SQLCODE = cccc.

**Explanation:** DB2 did not correctly prepare the statement because a table name or similar correlation was used as the high-level qualifier. Although valid in SQL, this does not work under a prepare of SQL statements.

User response: Remove the offending high-level qualifier and try the query again.

---

**ANL1032I**  PRECISE YES SELECTED BUT USER HAS NO DSN_STATEMNT_TABLE UNDER authid. NORMAL COSTING WILL BE EMPLOYED.

**Explanation:** You have requested to use the optimizer’s CPU and Service Unit estimates with Precise Yes parameter, but the required explain table is not available to obtain the information. SQL PA might attempt to create a DSN_STATEMNT_TABLE. Otherwise, it reverts to Precise No costing formulas.

User response: Build generic DSN_STATEMNT_TABLE entries for general use or refer to an existing table using the USEPLAN parameter to identify the authid.

---

**ANL1033W**  DB2 CANNOT PREPARE AN UPDATE OR DELETE WHERE CURRENT OF CSR BECAUSE CURSOR IS UNKNOWN. PGM WILL HANDLE SITUATION AND REPROCESS STATEMENT.

**Explanation:** An UPDATE or DELETE statement was processed by SQL PA that contained the WHERE CURRENT OF CURSOR clause. Because SQL PA is preparing the statement independently, there is no Declare Cursor to reference. SQL PA automatically handles the situation internally, adjusting for the update or delete in place. SQL code is -504.

User response: None.

---

**ANL1034E**  PROCESSING ERROR ENCOUNTERED DURING SQL STATEMENT ANALYSIS. ANLWORK TEMPORARY WORKFILE WAS NOT DEFINED.

**Explanation:** One of the files required for processing was not properly allocated. ANLWORK holds the SQL extracted from multiple DBRM members.

User response: Check for an ANLWORK DD card in batch JCL. For all other environments, contact IBM Software Support.

---

**ANL1035E**  UNABLE TO PROCESS MULTIPLE DBRM MEMBERS BECAUSE ANLIN PDS COULD NOT BE OPENED SUCCESSFULLY. RETURN CODE = rc, REASON = rs, INFO = in.

**Explanation:** The DBRM library named ddname (filename) ANLIN could not be opened by the BPAM access routine.

User response: Check the spelling to verify that you have specified the data set name correctly. Attempt to access this data set using the ISPF Edit or Browse to ensure the directory is intact. If this does not resolve the problem, provide the return code, reason, and info code to the IBM Software Support Center.

---

**ANL1039E**  UNABLE TO PROCESS MULTIPLE DBRM MEMBERS BECAUSE DSERV (LIST PDS DIRECTORY) FAILED. RETURN CODE = rc, REASON = rs, INFO = in.

**Explanation:** The directory of the DBRM library named ddname (filename) ANLIN could not be read by the BPAM access routine.

User response: Check that this directory is not corrupted by using the TSO LISTDS data set ME command, or use ISPF Edit or Browse to ensure the directory is intact. If this does not resolve the problem, provide the return code, reason, and info code to the IBM Software Support.

---

**ANL1040E**  DIRECTORY MEMBERS EXCEED THE LIMIT OF 2730—ONLY THE FIRST 2730 MEMBERS WILL BE PROCESSED. RETURN CODE = rc, REASON = rs, INFO = in.

**Explanation:** The DBRMKEY pattern has returned a match on more than 2730 members, the limit for a single run. The first 2730 members are handled.

User response: Restate the DBRMKEY pattern so that
less than 2730 members are selected, or run several jobs to pick up all of the required members by modifying the pattern.

**ANL1041E \* ANL1048W**

**ANL1041E \* UNABLE TO PROCESS MEMBER**

*member-name*, *THIS NAME WAS NOT FOUND IN THE DIRECTORY.\*

*RETURN CODE = rc, REASON = rs, INFO = in.*

**Explanation:** The member name specified for this DBRM library could not be located in the directory by the BPAM access routine.

**User response:** Check the spelling to verify that you have specified the member name correctly. Attempt to access this data set by using ISPF Edit or Browse to ensure the directory is intact. If this does not resolve the problem, provide the return code, reason, and info code to the IBM Software Support.

**ANL1043E \* DBRM MEMBER memname HAD A READ ERROR AND CANNOT BE PROCESSED. RETURN CODE = rc, REASON = rs, INFO = in.*

**Explanation:** The member name specified for this DBRM library could not be read by the BPAM access routine.

**User response:** Check the spelling to verify that you have specified the member name correctly. Attempt to access this data set by using ISPF Edit or Browse to ensure the directory is intact. If this does not resolve the problem, provide the return code, reason, and info code to the IBM Software Support.

**ANL1044E \* NO MEMBERS FOUND TO MATCH DBRMKEY PATTERN pattern. PROGRAM TERMINATES.**

**Explanation:** The pattern specified, with or without wild cards, for the DBRMKEY parameter did not match any members in the PDS directory. Valid pattern combinations are:

- (8 blanks) for all members
- * for all members
- memberx for a specific member
- ABC* for all member names starting with ABC
- %%%ABC%% for all members with any characters in the % positions and the characters A, B, and C in positions 4, 5, and 6

When a pattern uses the % wild card, members are selected only if the member name is the exact length of the pattern.

**User response:** Specify a more precise pattern in the DBRMKEY parameter and run the job again.

**ANL1045W \* USER SPECIFIED USEPLAN BUT INSTALLATION HAS NOT AUTHORIZED OPTION. SQL PA WILL SEARCH REGISTRY AND/OR CURRENT SQLID FOR ALTERNATIVES.**

**Explanation:** You attempted to use USEPLAN authid in ANLPARM or under TSO or Batch, but the installation has not authorized its use with a corresponding SETPLAN YES in the ANLCNTL parameters. Normal search for generic or user PLAN_TABLE continues.

**User response:** Enable this parameter with SETPLAN YES.

**ANL1046I \* USER HAS SPECIFIED USE OF authid.PLAN_TABLE FOR THIS RUN.**

**Explanation:** Successful implementation of a USEPLAN authid specification, that was authorized by the installation with SETPLAN YES. The PLAN_TABLE, and its owning authid, are displayed for your information.

**User response:** None.

**ANL1047W \* UNABLE TO USE authid.PLAN_TABLE, SQLCODE = ccc. SQL PA WILL SEARCH REGISTRY AND/OR CURRENT SQLID FOR ALTERNATIVES.**

**Explanation:** The PLAN_TABLE that was selected with SETPLAN authid was not found and cannot be allocated by the program. The likely SQLCODE is -518. SQL PA uses the standard alternatives starting with the Registry.

**User response:** Check for, or create, the PLAN_TABLE under the authid named in the USEPLAN. Run the job again.

**ANL1048W \* POSSIBLE AUTHORIZATION PROBLEMS WITH PLAN_TABLE USED.**

**Explanation:** When message ANL2012W occurs, and there were actually one or more good statements processed by SQL PA, this message is issued, indicating a likely problem with RACF\* authorizations in reading and writing the chosen authid.PLAN_TABLE.

**User response:** Check the JES log in batch for ICH408I messages, and their DB2 access privileges on the selected table.
**ANL1049W** RESOURCE LIMIT FACILITY HAS REJECTED QUERY qqqqqqqqq, SQLCODE = ccccc. SET RLFASUERR AND RLFASUWARN TO HIGHER VALUES FOR THIS AUTHID. STATEMENT CANNOT BE EVALUATED BY SQL PA.

**Explanation:** The Resource Limit Facility (RLF) has set CPU constraints too low to process a Prepare or Explain of this SQL statement under this authorization ID.

**User response:** Modify the values for RLFASUERR and RLFASUWARN as indicated in the DSNRLST table to increase the amount of CPU time available before RLF terminates the Prepare or Explain query with +/- 495 SQLCODE.

**ANL1050E** A CHAR-NUMERIC CONVERSION ERROR HAS OCCURRED DURING I/O.

**Explanation:** An invalid format was encountered on a read or write to a file.

**User response:** Check DCB information for all files allocated. If DCB is current, then contact IBM Software Support.

**ANL1051W** UNABLE TO CREATE user.DSN_STATEMNT_TABLE FOR PRECISE YES REQUEST. PGM CONTINUES PROCESSING WITHOUT THIS INFORMATION. RETCODE = rrr.

**Explanation:** SQL PA attempted to create a DSN_STATEMNT_TABLE dynamically because none existed and you requested PRECISE YES. However, it was not able to create the table.

**User response:** None. The program continues processing using normal costing options (Precise No).

**ANL1052W** UNABLE TO CREATE user.DSN_FUNCTION_TABLE FOR PRECISE YES REQUEST. PGM CONTINUES PROCESSING WITHOUT THIS INFORMATION. RETCODE = rrr.

**Explanation:** SQL PA attempted to create a DSN_FUNCTION_TABLE dynamically because none existed and you requested PRECISE YES or ALL. However, it was not able to create the table.

**User response:** None. The program continues processing using normal costing options (Precise No).

**ANL1053W** UNABLE TO CREATE user.DSN_DETCOST_TABLE FOR PRECISE ALL REQUEST. PGM CONTINUES PROCESSING WITHOUT THIS INFORMATION. RETCODE = rrr.

**Explanation:** SQL PA attempted to create a DSN_DETCOST_TABLE dynamically because none existed and you requested PRECISE ALL. However, it was not able to create the table.

**User response:** None. The program continues processing using normal costing options (Precise No).

**ANL1054I** PRECISE YES SELECTED BUT USER HAS NO DSN_FUNCTION_TABLE UNDER authid. FUNCTIONS WILL BE OMITTED.

**Explanation:** SQL PA cannot find or create a DSN_FUNCTION_TABLE and you requested PRECISE YES or ALL. Function analysis and reporting are omitted.

**User response:** None. The program continues processing without this additional information.

**ANL1055I** PRECISE ALL SELECTED BUT USER HAS NO DSN_DETCOST_TABLE UNDER authid. NORMAL COSTING WILL BE EMPLOYED.

**Explanation:** SQL PA cannot find or create a DSN_DETCOST_TABLE and you requested PRECISE ALL. Additional precision, analysis, and reporting are omitted.

**User response:** None. The program continues processing without this additional information.

**ANL1056W** ERROR PROCESSING user.DSN_DETCOST_TABLE, SQLCODE = ccccc. PGM IGNORES THIS TABLE AND CONTINUES PROCESSING QUERIES. QUERY NOT FOUND WAS: qqqqqqqqqqq, PLAN CLOUNT = ppppp.

**Explanation:** SQL PA did not find an expected entry in DSN_DETCOST_TABLE, yielding SQLCODE ccccc during PRECISE ALL processing.

**User response:** None. The program continues processing using normal costing options (Precise No).

**ANL1057W** ERROR PROCESSING user.DSN_FUNCTION_TABLE, SQLCODE = ccccc. PGM IGNORES THIS TABLE AND CONTINUES PROCESSING QUERIES. QUERY NOT FOUND WAS: qqqqqqqqqqq, PLAN CLOUNT = ppppp.
Explanation: SQL PA did not find an expected entry in DSN_FUNCTION_TABLE, yielding SQLCODE ccccc during PRECISE YES|ALL processing. In Version 8, a zparm (SPRMXPL) must be activated to turn on function table writes for built-in functions; however, UDFs and STPs are written to this table, regardless of DB2 release or zparm setting.

User response: None. The program continues processing using normal costing options (Precise No).

ANL1058E SQL PA COULD NOT DROP user.DSN_STATEMNT_TABLE, RETURN CODE = ccccc.

Explanation: SQL PA created a DSN_STATEMNT_TABLE for this run but was unable to delete it after processing was complete.

User response: Examine the return code (SQLCODE) and determine the cause of this problem.

ANL1059E SQL PA COULD NOT DROP user.DSN_FUNCTION_TABLE, RETURN CODE = ccccc.

Explanation: SQL PA created a DSN_FUNCTION_TABLE for this run but was unable to delete it after processing was complete.

User response: Examine the return code (SQLCODE) and determine the cause of this problem.

ANL1060E SQL PA COULD NOT DROP user.DSN_DETCOST_TABLE, RETURN CODE = ccccc.

Explanation: SQL PA created a DSN_DETCOST_TABLE for this run but was unable to delete it after processing was complete.

User response: Examine the return code (SQLCODE) and determine the cause of this problem.

ANL1061I TABLEAU BUILD ON tcode RESULTS IN RC rrrrr.

Explanation: SQL PA has attempted to drop an explain table that was created dynamically. The associated return code provides the status of that operation (0 was successful). The tcode values are: PLN = PLAN_TABLE STM = DSN_STATEMNT_TABLE FUN = DSN_FUNCTION_TABLE CST = DSN_DETCOST_TABLE PRD = DSN_PREDICAT_TABLE

User response: Examine the return code (SQLCODE), if it is non-zero, and determine the cause of the problem.

ANL1063E UNICODE DBRM CONVERSION ERROR, RC = rrrrr, REASON = cccccc.

Explanation: A Version 8 DBRM module could not be converted from Unicode to EBCDIC. The RC and REASON are return code and reason codes from the MVS conversion routine.

User response: The program cannot process this DBRM. Contact IBM Software Support with return and reason codes for analysis.

ANL1064W STRING TRUNCATION HAS OCCURRED FOR MIXED DATA (SBCS, DBCS). SQL PA WILL CONTINUE PROCESSING DATA, COND = cccc.

Explanation: A mixed data variable or field name was truncated by the program because it exceeded the size of the storage area it was being translated into.

User response: The program continues processing. If the SQL is not properly explained, check the spelling of the mixed data name, or substitute a shorter value.

ANL1065W A NON-CRITICAL UNDERFLOW ERROR HAS OCCURRED DURING PROCEDURES. SQL PA WILL CONTINUE PROCESSING DATA.

Explanation: An internal math underflow error has occurred while manipulating internal variables in the program, which does not affect the outcome of path lengths or cost estimates.

User response: None. The program continues processing.

ANL1066I PGM CANNOT ESTIMATE COST OF DECLARED GLOBAL TEMPORARY TABLE, SINCE NO DATA EXISTS IN THE CATALOG.

Explanation: The PREPARE failed because DB2 could not find catalog data describing the table, which was a
declared global temporary table, carrying the name SESSIONtablename. The table is created during SQL operations, and only exists for the life of the application. Thus, DB2 cannot find the permanent table during PREPARE, and issues SQLCODE -204: table not found.

User response: None. This message is informational. There are no statistics in the catalog for a declared global temporary table. DB2 cannot PREPARE the statement, and therefore SQL PA cannot evaluate it.

ANL1067W  UNABLE TO CREATE user.DSN_PREDICAT_TABLE FOR PRECISE ALL REQUEST. PGM CONTINUES PROCESSING WITHOUT THIS INFORMATION. RETCODE = rrr.

Explanation: SQL PA attempted to create a DSN_PREDICAT_TABLE dynamically because none existed and you requested PRECISE ALL. However, it was not able to create the table.

User response: None. The program continues processing using normal costing options (PRECISE NO).

ANL1068I  ERROR PROCESSING user.DSN_PREDICAT_TABLE, SQLCODE = cccc. PGM IGNORES THIS TABLE AND CONTINUES PROCESSING QUERIES. QUERY NOT FOUND WAS: uestioned, PLAN CLOUNT = ppppp.

Explanation: SQL PA did not find an expected entry in DSN_PREDICAT_TABLE, yielding SQLCODE cccc during PRECISE ALL processing.

User response: None. The program continues processing using normal costing options (PRECISE NO).

ANL1069E  SQL PA COULD NOT DROP user.DSN_PREDICAT_TABLE. RETCODE = trc.

Explanation: SQL PA dynamically created a plan table under the current authid for this operation, but was unable to drop the table at the conclusion of the run. The value of trc is the SQLCODE returned from the DROP table attempt.

User response: Either a system or program error has occurred or you do not have permission to drop tables on this system. Check SQLCODE for further assistance.

ANL1072E  SQL PA COULD NOT DROP authid.PLAN_TABLE. RETCODE = trc.

Explanation: SQL PA found no usable plan table under current authid, and no generic plan tables were available. It attempted to create one dynamically but was unable to do so. The value of trc is the SQLCODE returned from the CREATE attempt.

User response: Either you have not been granted use of the generic SQL PA table space and database, it does not exist, or you do not have permission to create tables on this system. Check SQLCODE for further assistance. Message ANL1008E follows with further information.

ANL1073W  "EXPLAIN ANLEEE" BUT NO "EEECALL EEPATH" WAS FOUND. EEPATH INPUT ASSUMED FOR THIS RUN. ANLIN MUST BE "ANLEEE.SQL".

Explanation: The EXPLAIN ANLEEE parameter was set, but you did not set the corresponding EECALL EEPATH parameter. SQL PA assumes that this is what you wanted and the program continues, expecting that data is found in the EEPATH table and in the ANLEEE.SQL input file.

User response: Code the EECALL EEPATH parameter along with EXPLAIN ANLEEE.

ANL1074E  "EXPLAIN [EEE|EXT]" WAS SPECIFIED BUT ANLIN IS NOT "ANLEEE.SQL" (EEE).

Explanation: ANLIN file must have been created by EEE to use the [EEE|EXT] option. EXPLAIN NEW is assumed, and SQL PA processing continues.

User response: None.
User response: None.

Full message text:

TOTAL NUMBER OF PLAN RECORDS PROCESSED WAS nnn
TOTAL NUMBER OF PARSED RECORDS SCANNED WAS mmm
TOTAL NUMBER OF ERRORS ENCOUNTERED IN RUN mmm
CATALOG READS – STORED:
   TABLES = n  TABSPACES = m
   INDEXES = l  ROUTINES = u
   IDXKEYS = k  COLUMNS = c
CAT READS – NOT STORED:
   RELATIONS = r
   COLDIST = d  ALT INDEX = a

ANL2001E  FATAL SQL ERROR ENCOUNTERED DURING EXPLAIN/CATALOG ACCESS. DB2 SQLCODE = cccc. DB2 OPERATION: objname, @ EXPLAIN REC: nn, @ PARSER REC: mmm.

Explanation: An SQL error occurred during processing. The current DB2 object being accessed is indicated, and where the processing error occurred in the plan table and parser input files.

User response: Check plan table and input files for errors. If the error continues, contact IBM Software Support.

Note: Depending upon the error, this message can also indicate a Possible Cause for the condition and suggest hints for resolution:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Possible Cause/Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>Record length is wrong size. Check DCB information.</td>
</tr>
<tr>
<td>42</td>
<td>Transmission error occurred reading an input file, due to missing file DD, no member name for PDS, and so on.</td>
</tr>
<tr>
<td>83</td>
<td>DCB information incorrect - check LRECL, RECFM, BLKSZ.</td>
</tr>
<tr>
<td>84</td>
<td>No DD statement for required data set – missing DD.</td>
</tr>
<tr>
<td>612</td>
<td>613</td>
</tr>
</tbody>
</table>
numeric, or Steplib to DSNLOAD or SCEERUN are required.

1040
Output file used all available space – define larger.

3920
Program out of storage – run with larger region size.

8094
Protection Exception – check authorization ID/RACF permissions.

8097
Data Check on input — DBRM was expected by program.

9950
Input data set or PDS member was not found for ANLIN or internal PL/I abend.

An internal error occurred during processing. Save the error code, DB2 object, and position in the plan table and save the parser input file for DB2 Support.

User response: Check plan table and input files for errors. Explore possible causes if reported. If the error continues, contact IBM Software Support.


Explanation: The plan table and parser input file are out of synchronization due to an error in parsing or a transformation of the query undetected by SQL PA. The current query number and block number of each are displayed, and record count and position in each file.

User response: If the DBRM statement numbers are out of sequence, then reprocess with the PROCESS NOSEQ parameter set. Check plan table and input files for errors. If the error continues, note EXPLAIN and PARSER record information and contact IBM Software Support.


Explanation: Informational. An ALIAS was uncovered. The correct table name was substituted for the catalog statistics.

User response: None.


Explanation: Informational. This query materializes a view, as indicated by view name.

User response: None.

ANL2006I NO TABLE STATS FOR QUERY mmmmmmmmm QBLOCK mnn. DEFAULTS USED.

Explanation: Informational. No statistics were found in the catalog for the table referenced by this query block.

User response: None.

ANL2007I NO TSPACE STATS FOR QUERY mmmmmmmmm QBLOCK mnn. DEFAULTS USED.

Explanation: Informational. No statistics were found in the catalog for table space referenced by this query block.

User response: None.

ANL2008I NO INDEX STATS FOR QUERY mmmmmmmmm QBLOCK mnn. DEFAULTS USED.

Explanation: Informational. No statistics were found in the catalog for the index referenced by this query block.

User response: None.

ANL2009I OPTIMIZER HAS TRANSFORMED QUERY: mmmmmmmmm. WAS A SUBQUERY... NOW A JOIN, FOR YOUR INFORMATION.

Explanation: Informational. SQL PA has detected an optimizer transformation from a subquery to a join.

User response: None.

ANL2010I OPTIMIZER HAS TRANSFORMED QUERY: mmmmmmmmm. WAS A JOIN... NOW A SUBQUERY, FOR YOUR INFORMATION.

Explanation: Informational. SQL PA has detected an optimizer transformation from a join to a subquery.

User response: None.
| ANL2011I | OPTIMIZER HAS TRANSFORMED
_QUERY: mmmmmmmmmmm. WAS A
SUBQUERY... NOW A PRED, FOR
YOUR INFORMATION. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Informational. SQL PA has detected a transformation from a subquery to a simple predicate.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None.</td>
</tr>
</tbody>
</table>

| ANL2012W | NO VALID PLANS WERE
GENERATED BY INPUT SQL. (CHECK
CATALOG FOR OBJECTS EXISTENCE & QUERY SYNTAX). |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>SQL PA was unable to Prepare or Explain any of the SQL input, indicating that the objects referred to in the query did not exist in the catalog, or the SQL had syntax errors or could not be prepared.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check for these conditions, correct, and run again.</td>
</tr>
</tbody>
</table>

| ANL2013I | GLOBAL TEMPORARY TABLE ON
_QUERY: mmmmmmmmm QBLOCK: mnn.
GLOBAL NAME USED IS: tempname. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Informational. This query materializes a global temporary table, as indicated by table name tempname.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None.</td>
</tr>
</tbody>
</table>

| ANL2014I | AUXILIARY TABLE (LOBs) ON QUERY:
mmmmmmmm QBLOCK: mnn.
AUX TABLE NAME USED IS: tempname. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Informational. This query uses an auxiliary table to hold LOBs, as indicated by table name tempname.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None.</td>
</tr>
</tbody>
</table>

| ANL2015I | NO KEY STATS FOR QUERY:
BLOCK: qqqq INDEX KEY NOT
FOUND: xxxyyyyy. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Informational. Statistics are unavailable during processing.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None.</td>
</tr>
</tbody>
</table>

| ANL2016I | NO COLUMN STATS FOR QUERY:
BLOCK: qqqq COLUMN NOT
FOUND: cccccccccc. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Informational. Statistics are unavailable during processing.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None.</td>
</tr>
</tbody>
</table>

| ANL2017I | NO DISTRIBUTION STATS FOR
_QUERY: mmmmmmm QBLOCK: qqqq.
COLUMN NOT FOUND: cccccccccc. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Informational. Statistics are unavailable during processing.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None.</td>
</tr>
</tbody>
</table>

| ANL2018I | NO ROOM FOR CHOSEN INDEX,
_QUERY: qqqq QBLOCK: bbb MAX
1024 EXCEEDED. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The internal array size for storing information about the indexes chosen for access paths has been exceeded. There is room for up to 1024 indexes.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Reduce the amount of SQL being processed in a single run of SQL PA.</td>
</tr>
</tbody>
</table>

| ANL2019I | NO ROOM FOR ALTERNATE INDEX,
_QUERY: qqqq QBLOCK: bbb MAX
1024 EXCEEDED. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The internal array size for storing information about the other indexes not chosen for access paths has been exceeded. There is room for up to 1024 indexes.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Reduce the amount of SQL being processed in a single run of SQL PA.</td>
</tr>
</tbody>
</table>

| ANL2020I | NO ROOM FOR ALT IDX KEYS,
_QUERY: qqqq QBLOCK: bbb MAX
4096 EXCEEDED. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The internal array size for storing information about the alternative index keys has been exceeded. There is room for up to 4096 index key entries.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Reduce the amount of SQL being processed in a single run of SQL PA.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL2021W</th>
<th>The VERSION is set to vvvv, but yyyy was detected and used. Update the VERSION parameter.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The VERSION parameter was set for one version of DB2 (vvvv), but SQL PA detected and used a different version (yyyy) for analysis.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>If you want to use version vvvv for cost analysis, then reprocess with the parameter PROCESS ASIS set. Correct the VERSION parameter to reflect the current operating release.</td>
</tr>
</tbody>
</table>
ANL2022I NO TABLE STATS FOR QUERY: qqqqqqqq BLOCK: bbb JOIN WORK TABLE: explan.tname.

Explanation: The DSNWFQBnn temporary join work file was used in this plan step.

User response: None. This message avoids the redundant no stats for table message.

ANL2027I NO TABLE STATS FOR QUERY: qeryno BLOCK: qblockno COMMON TABLE EXPRESSION OR RECURSIVE COMMON TABLE EXPRESSION WAS USED.

Explanation: SQL of the form WITH table-expression AS table name, SELECT... was used to define a table and then access it in subsequent SQL. This type of SQL is referred to as a common table expression, and it allows you to define an instant table or view of the data, and then use it, within a single SQL statement.

User response: None. This message is informational. There are no stats in the catalog for an instantly defined table.

ANL2028W NO PRIOR ACCESS PLAN WAS FOUND IN EEepath FOR QUERYNO nnnnnnnn. OLD PLAN AND COMPARISON TO NEW PLAN ARE NOT AVAILABLE. PROGRAM CONTINUES WITH NEXT QUERY.

Explanation: The QUERYNO requested could not be found in the EEepath table. Program proceeds to the next statement.

User response: Run ANLEEE to collect data in EEepath using the TOPA=ALL parameter for the required QUERYNO and run SQL PA again.

ANL2029E ERROR OCCURRED ACCESSING EEepath TABLE, SQLCODE = sqlcode. OLD PLAN AND COMPARISON TO NEW PLAN ARE NOT AVAILABLE. PROGRAM CONTINUES WITH NEXT QUERY.

Explanation: A processing error occurred while accessing the EEepath table.

User response: Check the SQLCODE and correct the problem, or run ANLEEE maintenance to clean up the EEepath table and re-populate the data.

ANL2030I NO HISTOGRAM DISTRIBUTION STATS FOR QUERY: qeryno BLOCK: qblockno BUFFERS IN MEMORY USED TO PROCESS THIS STATEMENT.

Explanation: The access path used buffered data in memory for this materialized table access, created from a related SQL operation. No I/O was necessary to read the table pages, which were already stored in the buffer pool.

User response: None. This message is informational. There are no stats in the catalog for a virtual buffered table.

ANL3000E UNABLE TO ACCESS STATISTICS ARRAY: ASTAT. SQL PA TERMINATES WITH UNRECOVERABLE DATA ERROR.

Explanation: An array used to hold catalog statistics has been corrupted during processing.
User response: Run ANLDEBUG JCL with diagnostics turned on (add a DBTRACE DMP parameter at the top of the ANLCNTL file) and contact IBM Software Support.

**ANL3001E**  UNABLE TO ACCESS PARSER ARRAY: APAS. SQL PA TERMINATES WITH UNRECOVERABLE DATA ERROR.

Explanation: An array used to hold parser intelligence was corrupted during processing.

User response: Run ANLDEBUG JCL with diagnostics turned on (add a DBTRACE DMP parameter at the top of the ANLCNTL file) and contact IBM Software Support.

**ANL3002E**  UNABLE TO ACCESS THE EXPLAIN ARRAY: AEXP. SQL PA TERMINATES WITH UNRECOVERABLE DATA ERROR.

Explanation: An array used to hold explain plan data has been corrupted during processing.

User response: Run ANLDEBUG JCL with diagnostics turned on (add a DBTRACE DMP parameter at the top of the ANLCNTL file) and contact IBM Software Support.

**ANL3003W**  UNABLE TO OPEN REPORTS FILE ANLREP. SQL PA WILL NOT PROVIDE ENHANCED EXPLAIN REPORT.

Explanation: The output file identified by ANLREP DD is not allocated or is unusable. The enhanced explain report cannot be created.

User response: Correct the problem with ANLREP DD and run again.

**ANL3004W**  UNABLE TO OPEN REPORTS FILE QTRACE. SQL PA WILL NOT PROVIDE A DETAILED TRACE REPORT.

Explanation: The output file identified by QTRACE DD is not allocated or is unusable. The detailed trace report cannot be created.

User response: Correct the problem with QTRACE DD and run again.

**ANL3005W**  VERSION vvvv IS NOT SUPPORTED BY THIS PROGRAM: yyyyy ASSUMED.

Explanation: The version selected by VERSION parameter (vvvv) is not supported by this release of the product. The default version (yyyy) is substituted.

User response: None. SQL PA supports three versions of DB2 simultaneously, the current and two back levels.

**ANL3006W**  DEVICE dddddd IS NOT SUPPORTED BY THIS PROGRAM: 3390-2 ASSUMED.

Explanation: An unknown device type was entered in the user parameter file. SQL PA uses 3390-3 device.

User response: Correct the STORAGE parameter in the user parameter file. Optionally, define your own device using the NEWSTOR parameter and related parameters.

**ANL3007E**  UNABLE TO ACCESS PARSER DATA ANLPAS. SQL PA TERMINATES WITH AN UNRECOVERABLE ERROR.

Explanation: An array used to hold parser intelligence cannot be accessed.

User response: Run ANLDEBUG JCL with diagnostics turned on (add a DBTRACE DMP parameter at the top of the ANLCNTL file) and contact IBM Software Support.

**ANL3008E**  UNABLE TO ACCESS STATISTICS IN ANLSEP. SQL PA TERMINATES WITH AN UNRECOVERABLE ERROR.

Explanation: An array used to hold access plan data cannot be accessed.

User response: Run ANLDEBUG JCL with diagnostics turned on (add a DBTRACE DMP parameter at the top of the ANLCNTL file) and contact IBM Software Support.

**ANL3009E**  SYNCHRONIZATION PROBLEM BETWEEN DATA INPUTS: EXPLAIN QUERY IS qqueryno qblkno planno mixopseq PARSER QUERY IS qqueryno qblkno RELATION QUERY IS qqueryno qblkno SQL PA PROGRAM TERMINATES WITH UNRECOVERABLE ERROR.

Explanation: The cost estimation process finds inconsistencies between several inputs representing the PLAN_TABLE, parser intelligence, and relationships. SQL PA cannot continue. Queryno is the current query number and qblkno is the current query block for the respective input sources. Planno and Mixopseq are plan step and multiple index steps respectively from the access plan.

User response: Run ANLDEBUG JCL with diagnostics turned on (add a DBTRACE DMP parameter at the top of the ANLCNTL file) and contact IBM Software Support.
ANL3010E • ANL3019I

ANL3010E UNABLE TO OPEN STATEMENTS FILE ANLOUT. SQL PA TERMINATES WITH UNRECOVERABLE FILE ERROR.

Explanation: The file identified in the ANLOUT DD statement cannot be opened.

User response: Correct errors in the DD statement and run again.

ANL3011E UNABLE TO READ STATEMENTS FILE ANLOUT. SQL PA TERMINATES WITH UNRECOVERABLE FILE ERROR.

Explanation: The file identified by ANLOUT DD is not allocated, is unusable, or is corrupted. This file holds the SQL statements for report presentation.

User response: Correct the problem with ANLOUT DD and run again. If the problem persists, contact IBM Software Support.

ANL3012W UNABLE TO CREATE REPORTS FILE QLIMIT. SQL PA WILL NOT PROVIDE A QUERY LIMITS REPORT.

Explanation: The output file identified by QLIMIT DD is not allocated or is unusable. The query limits report cannot be created.

User response: Correct the problem with QLIMIT DD and run again.

ANL3013W UNABLE TO ACCESS RELATIONS ARRAY: ARELS. SQL PA WILL NOT SHOW RI INFORMATION FOR THIS RUN.

Explanation: An array used to hold referential integrity data cannot be accessed.

User response: The program runs without providing RI relationship data. If the problem persists, run ANLDEBUG JCL with diagnostics turned on (add a DBTRACE DMP parameter at the top of the ANLCNTL file) and contact IBM Software Support.

ANL3014W UNABLE TO ACCESS RELATIONS IN ANLREL. SQL PA WILL NOT DISPLAY ANY FURTHER RI INFO.

Explanation: An array used to hold referential integrity data cannot be accessed during the program.

User response: The program runs without providing RI relationship data from this point forward. If the problem persists, run ANLDEBUG JCL with diagnostics turned on (add a DBTRACE DMP parameter at the top of the ANLCNTL file) and contact IBM Software Support.

ANL3015W AN UNKNOWN PARAMETER WAS READ FROM THE ANLPARM FILE. THE KEYWORD SCANNED WAS: xxxxxx. THIS PARM WAS IGNORED.

Explanation: A parameter read from ANLPARM is not recognized as valid. SQL PA ignores the parameter.

User response: Correct the parameter spelling and run again.

ANL3016W AN UNKNOWN PARAMETER WAS READ FROM THE ANLCNTL FILE. THE KEYWORD SCANNED WAS: xxxxxx. THIS PARM WAS IGNORED.

Explanation: A parameter read from ANLCNTL is not recognized as valid. SQL PA ignores the parameter.

User response: Correct the parameter spelling and run again.

ANL3017I NO STATISTIC FOR CARDF ON TABLE authid.tablename OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnnn.

Explanation: The DB2 catalog contained a default (usually -1) for this statistic for the table authid.tablename referred to by query nnnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object.

User response: None. Optionally, run RUNSTATS or populate the catalog with the statistics you want, and run again.

ANL3018I NO STATISTIC FOR NPAGES|F ON TABLE authid.tablename OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnnn.

Explanation: The DB2 catalog contained a default (usually -1) for this statistic for the table authid.tablename referred to by query nnnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object. Depending on the level of DB2, the statistic is NPAGES or NPAGESF.

User response: None. Optionally, run RUNSTATS or populate the catalog with the statistics you want, and run again.

ANL3019I NO STATISTIC FOR PCTPAGES ON TABLE authid.tablename OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnnn.

Explanation: The DB2 catalog contained a default (usually -1) for this statistic for the table authid.tablename referred to by query nnnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object.
User response: None. Optionally, run RUNSTATS or populate the catalog with the statistics you want, and run again.

**ANL3020I** NO STATISTIC FOR NACTIVEF ON TABLE authid.tablename OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnn.

Explanation: The DB2 catalog contained a default (usually 0) for this statistic for the table authid.tablename referred to by query nnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object. Depending on the level of DB2, the statistic is NACTIVE or NACTIVEF.

User response: None. Optionally, run RUNSTATS or populate the catalog with the statistics you want, and run again.

**ANL3021I** NO STATISTIC FOR NLEAF ON INDEX authid.indexname OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnn.

Explanation: The DB2 catalog contained a default (usually -1) for this statistic for the table authid.tablename referred to by query nnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object.

User response: None. Optionally, run RUNSTATS or populate the catalog with the statistics you want, and run again.

**ANL3022I** NO STATISTIC FOR NLEVELS ON INDEX authid.indexname OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnn.

Explanation: The DB2 catalog contained a default (usually -1) for this statistic for the table authid.tablename referred to by query nnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object.

User response: None. Optionally, run RUNSTATS or populate the catalog with the statistics you want, and run again.

**ANL3023I** NO STATISTIC FOR FIRSTKEYCARDF ON INDEX authid.indexname OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnn.

Explanation: The DB2 catalog contained a default (usually -1) for this statistic for the table authid.tablename referred to by query nnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object.

User response: None. Optionally, run RUNSTATS or populate the catalog with the statistics you want, and run again.

**ANL3024I** NO STATISTIC FOR FULLKEYCARDF ON INDEX authid.indexname OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnn.

Explanation: The DB2 catalog contained a default (usually -1) for this statistic for the table authid.tablename referred to by query nnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object.

User response: None. Optionally, run RUNSTATS or populate the catalog with the desired statistic, and run again.

**ANL3025I** NO STATISTIC FOR CLUSTERRATIOF ON INDEX authid.indexname OPTIMIZER DEFAULT USED FOR QUERYNO nnnnnnnn.

Explanation: The DB2 catalog contained a default (usually 0) for this statistic for the table authid.tablename referred to by query nnnnnnnn. SQL PA uses the same values that the DB2 optimizer uses internally for this object. Depending on the level of DB2, the statistic is CLUSTERRATIO or CLUSTERRATIOF.

User response: None. Optionally, run RUNSTATS or populate the catalog with the statistics you want, and run again.

**ANL3026W** NO STATISTICS FOUND IN CATALOG FOR ONE OR MORE VARIABLES. OPTIMIZER DEFAULTS WERE USED WHERE MISSING VALUES FOUND.

Explanation: Follow-up note in report files (ANLREP, QTRACE) indicating that some statistics are missing.

User response: None.

**ANL3027I** QUERYNO nnnnnn WAS TRANSFORMED BY THE DB2 OPTIMIZER.

Explanation: Informational note indicating a transformation was performed by the optimizer—the Plan does not match the original SQL. The costing is done for the new plan.

User response: None.
ANL3028E  THERE WERE NO STATISTICS COLLECTED FROM CATALOG FOR INPUT SQL. PGM CANNOT DETERMINE COST - LACKING CRITICAL DATA. MOST LIKELY, ANSQLPA WAS UNABLE TO PROCESS INPUT.

Explanation: This message indicates that the objects haven't been created yet. No statistics exist in the catalog for the objects referred to in the SQL. Defaults would have been be used if the objects exist, but RUNSTATS was not run.

User response: Create the objects, run RUNSTATS or populate the catalog with the statistics you want, and run the job again.

ANL3029W  THE EXPLAIN FOR QUERY IS NOT VALID. PGM CANNOT PROVIDE A COST ESTIMATE. STATEMENT IGNORED.

Explanation: No valid records where written to the plan table for this query. SQL PA cannot process the statement, and moves on to the next one.

User response: Correct any errors in the query and run again.

ANL3030E  CATALOG STATISTICS FOR THIS SQL HAVE CAUSED DIVISION BY ZERO. CHECK STATISTICS AND CORRECT, THEN RE-RUN PGM.

Explanation: One of the catalog statistics used by SQL PA for this statement has caused a division by zero in the path length calculations.

User response: Check the catalog statistics for the objects used, and run again.

ANL3031E  OVERFLOW ERROR DETECTED DURING CALCULATIONS. CATALOG STATISTICS ARE A LIKELY CAUSE. CHECK AND RE-RUN PGM.

Explanation: One of the catalog statistics used by SQL PA for this statement has caused a computational exception in the path length calculations.

User response: Check the catalog statistics for the objects used, and run again.

ANL3032E  UNDERFLOW ERROR DETECTED DURING CALCULATIONS. CATALOG STATISTICS ARE A LIKELY CAUSE. CHECK AND RE-RUN PGM.

Explanation: One of the catalog statistics used by SQL PA for this statement has caused a computational exception in the path length calculations.

User response: Check the catalog statistics for the objects used, and run again.

ANL3033E  FIXEDOVERFLOW ERROR DETECTED DURING CALCULATIONS. CATALOG STATISTICS ARE A LIKELY CAUSE. CHECK AND RE-RUN PGM.

Explanation: One of the catalog statistics used by SQL PA for this statement has caused a computational exception in the path length calculations.

User response: Check the catalog statistics for the objects used, and run again.

ANL3034E  SIZE ERROR DETECTED DURING CALCULATIONS. CATALOG STATISTICS ARE A LIKELY CAUSE. CHECK AND RE-RUN PGM.

Explanation: One of the catalog statistics used by SQL PA for this statement has caused a computational exception in the path length calculations.

User response: Check the catalog statistics for the objects used, and run again.

ANL3035I  THE ASSUMED OVERHEAD FOR CONNECTION TYPE OF connection-type HAS BEEN ESTIMATED IN MILLIONS OF INSTRUCTIONS FOR THESE ONLINE SQL STMTS: THREAD MGMT = ttttttttt ATTACH = aaaaaaa APPL = ppppppppp.

Explanation: Informational. SQL PA reports on the processor usage estimates used for the attach facility and related components.

User response: None.

ANL3036I  THE ASSUMED OVERHEAD FOR CONNECTION TYPE OF connection-type HAS BEEN ESTIMATED IN MILLIONS OF INSTRUCTIONS FOR THESE BATCH SQL STMTS: THREAD MGMT = ttttttttt ATTACH = aaaaaaa APPL = ppppppppp.

Explanation: Informational. SQL PA reports on the processor usage estimates used for the attach facility and related components.

User response: None.
ANL3037I  ESTIMATED storage DASD SERVICE TIMES ARE: SYNC READ 4K PAGE = syncrd PREFETCH BLOCK OF $nn$ 4K PAGES = prefrd SYNC READ 8K PAGE = sync8 PREFETCH BLOCK OF $nn$ 8K PAGES = pref8 SYNC READ 16K PAGE = sync16 PREFETCH BLOCK OF $nn$ 16K PAGES = pref16 SYNC READ 32K PAGE = sync32 PREFETCH BLOCK OF $nn$ 32K PAGES = pref32 ASYNC WRT 4K PAGE = async4 ASYNC WRITE OF 16K PAGE BLOCK = asyn16 ASYNC WRITE OF 8K PAGE = async8 ASYNC WRITE OF 32K PAGE BLOCK = async32

Explanation:  Informational. SQL PA computes the average DASD service times for all buffer pool page sizes based on the disk device types and specifications.

User response:  None.

ANL3038I  USER HAS DEFINED VALUES FOR THE newdasd STORAGE SUBSYSTEM.

Explanation:  Informational. Informs you that your own DASD device type has been defined and used for evaluation.

User response:  None.

ANL3039I  SHOWALT ACTIVE, BUT NO ALTERNATE INDEXES AVAILABLE FOR THIS ACCESS.

Explanation:  Informational. User-specified SHOWALT parameter, but this table does not have any alternative indexes to display.

User response:  None.

ANL3040W  UNABLE TO OPEN PARAMETER FILE parmfile

Explanation:  The parameter file identified by parmfile DD is not allocated or unusable. Default values are used for the corresponding parameters. The following values are the possible values for parmfile:

ANLCNTL  SQL PA system parameters

ANLPARM  SQL PA user parameters

DEFAULT  Plan table report parameters

User response:  Verify and correct the parameter files specified in the ANLPARM panel of the Change the DB2 SQL PA parameters option when modifying your configuration.

ANL3041I  NO EXPLAINABLE STATEMENTS FOUND IN FILENAME.

Explanation:  The specified location did not contain any explainable statements.

ANL3042I  PROCESSING SQL STATEMENTS FROM FILENAME.

Explanation:  SQL PA is processing the statements from the specified location.

ANL3099E  A TERMINAL PROCESSING ERROR HAS OCCURRED. PLI ERROR CODE =pppppp SQL PA ERROR CODE =cccccc RETURN CODE =rr.

Explanation:

Note: Depending upon the error, this message can also indicate a Possible Cause for the condition and suggest hints for resolution:

Error Code  Possible Cause/Resolution
22  Record length is wrong size. Check DCB information.
42  Transmission error occurred reading an input file, due to missing file DD, no member name for PDS, and so on.
83  DCB information incorrect - check LRECL, RECFM, BLKSZ.
84  No DD statement for required data set – missing DD.
340  Variable size error - check catalog stats, RUNSTATS.
612 | 613  Internal conversion error from character to numeric or Steplib to DSNLOAD or SCEERUN can be required.
1040  Output file used all available space – define larger.
3929  Program out of storage – run with larger region size.
8094  Protection Exception – check authorization ID/RACF permissions.
8097  Data Check on input — DBRM was expected by program.
Input data set or PDS member was not found for ANLIN or internal PL/I abend.

User response: Note the error and return codes and explore possible causes, if reported. Save any system dump. Contact IBM Software Support.

Explanation: A catastrophic error has occurred during ANLSQLPA processing that was not explained by any other messages.

ANL4001I SQLPA HAS CONNECTED TO SYSTEM AT location LEVEL version.

Explanation: Informational. You have requested to process this SQL at the remote location by including the VIAADRDA parameter in ANLPARMS, and SQL PA has successfully connected through the DRDA Connect statement to that system. The current version of DB2 running at that location is also returned as version (for example, DSN08011).

User response: None.

ANL4002I CONNECTION TO SYSTEM AT location HAS BEEN RELEASED.

Explanation: Informational. Your connection to the location has been released by a RELEASE statement, prior to terminating the local DB2 thread and session.

User response: None.

ANL4003I THE CAF OPEN RETURN CODE IS ccc FOR plan-name ON ssid.

Explanation: Informational. SQL PA uses the call attach facility to connect to DB2, and this shows the return code from the OPEN call (ccc), the plan name (plan-name) and the DB2 subsystem ID (ssid) that you are trying to connect to.

User response: If there are problems with the CAF Open, then check the SUBSYST parameter for correct DB2 subsystem name, and also see if it is started and available for access.

ANL4004I THE CAF CLOSE RETCODE IS ccc FOR plan-name ON ssid.

Explanation: Informational. SQL PA uses the call attach facility to connect to DB2, and this shows the return code from the CLOSE call (ccc), the plan name (plan-name) and the DB2 subsystem ID (ssid) that you are trying to disconnect from.

User response: If there are problems with the CAF CLOSE, check the SUBSYST parameter for the correct DB2 subsystem name, and see if it is started and available for access.

ANL4005W CONNECTION TO SYSTEM AT loc COULD NOT BE ESTABLISHED.

SQLPA WILL ATTEMPT TO PROCESS SQL ON THIS SUBSYSTEM (ssid).

SQLCODE RETURNED TO CALLER WAS sqlcode.

Explanation: Warning. The program attempted to CONNECT through DRDA to the system at loc but was unable to do so. Possible causes might be improper binding of the Remote Package, improper binding of the local plan (not including the correct PKLIST) or the DDF address space of the Remote facility might be down (this would generate a -30080 SQLCODE on the call). SQL PA attempts to run the SQL on the local processor, under the subsystem ID ssid. The SQLCODE returned by the error is also displayed.

User response: As appropriate.

ANL4006E CONNECTION TO SYSTEM location COULD NOT BE RELEASED.

SQLCODE = sqlcode.

Explanation: Error. After connecting through DRDA and processing SQL on the Remote server, the local program was unable to RELEASE the DRDA connection. This generally does not affect the outcome of processing (all work is done at this point and files stay intact) but investigate the cause of the error. Refer to SQLCODE for reasons.

User response: As appropriate.

S013 MVS ABEND - SYSTEM 013.

Explanation: When running SQL PA in batch mode, it is possible to encounter this error if the input data set exists but the member name does not. For example, a valid DBRM library was used for ANLIN, but the member name specified did not exist. This condition also results in JOB NOT RUN - JCL ERROR message. Under MVS, the member name is not checked until run time, therefore, the S013 abend.

User response: Check the spelling on the appropriate DD statement and run again.

U4091 LE ABEND.

Explanation: Unexpected condition occurred during LE processing with SQL PA. Possibly, a new release of LE was installed with other than default values for ERRCOUNT.

User response: Add override parameters to ANLATCH JCL:

//ANLSTEP1 EXEC PGM=ANLSQLPA,PARM=’ER(0)/’
//ANLSTEP2 EXEC PGM=ANLSQLPA,PARM=’ER(0)/’
Warnings and alerts

In general, the following messages alert you to special processing by SQL PA or warn you about characteristics of your query that might affect its performance. **Warnings** and **Alerts** are always available and are not affected by the ADVISOR parameter in the user parameters file (ANLPARM).

Where the message is self-explanatory, no additional explanation is given.

<table>
<thead>
<tr>
<th>ANL5001W</th>
<th>*** ALERT: USER HAS REQUESTED CLASS 1 CPU TIME ONLY IN COST ESTIMATES.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The ANLKEYS parameter in the configuration file has a value of 10000 or 12000.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None, if you want class 1 CPU time. Otherwise, select the correct configuration file and run again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL5002W</th>
<th>*** ALERT: THIS ABBREVIATED RUN ONLY EXERCISES THE PARSER; NO COSTING.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The ANLKEYS parameter in the configuration file has a value of 01000.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None, if you just want to parse the SQL. Otherwise, select the correct configuration file and run again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL5003W</th>
<th>*** ALERT: THIS SPECIAL RUN ELIMINATES OVERHEADS FOR CAPACITY PLANNING MODEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The ANLKEYS parameter in the configuration file has a value of 02000 or 12000.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None, if you want to eliminate overhead path lengths. Otherwise, select the correct configuration file and run again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL5004W</th>
<th>*** ALERT: PATH LENGTHS HAVE BEEN RECALIBRATED BY A FACTOR OF: KC.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>This message only occurs if the path lengths have been recalibrated by a special ANLKEYS parameter setting.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>None.</td>
</tr>
</tbody>
</table>

The rest of the warnings and alerts are related to your queries, not to SQL PA processing. You can change the SQL in your queries based on these messages or leave the SQL as it is. The choice is yours.
*** WARNING: THIS SQL STATEMENT CONTAINS NO PREDICATES AND USES NO BUILT-IN FUNCTIONS, SO DB2 IS SELECTING THE TABLESPACE SCAN ACCESS METHOD. IN GENERAL, THIS MEANS POOR PERFORMANCE, UNLESS THE TABLE IS VERY SMALL: VERIFY THE ACCURACY OF CATALOG STATISTICS (NPAGES, NACTIVE, CARD), AND REEVALUATE WHETHER THIS ACCESS PATH IS APPROPRIATE.

User response: As appropriate.

*** WARNING: THIS NONSEGMENTED TABLESPACE ACTUALLY CONTAINS MULTIPLE TABLES, MEANING THAT YOU ARE SCANNING UNNECESSARY PAGES THAT DO NOT HOLD ROWS FOR YOUR TABLE, INCLUDING PAGES IN THE TABLESPACE THAT BELONG TO OTHER TABLES NOT BEING ACCESSED HERE.

User response: As appropriate.

*** ALERT: THIS QUERY IS ACCESSING THE TABLE VIA A TABLESPACE SCAN AGAINST A TABLESPACE THAT IS NONSEGMENTED. EVEN THOUGH THIS IS JUST A SIMPLE TABLESPACE (ONLY ONE TABLE) YOU ARE NEEDLESSLY READING EXTRA PAGES BECAUSE DB2 MUST SCAN EVERY ACTIVE PAGE IN THE SPACE TO FIND ROWS.

User response: As appropriate.

*** ALERT: THIS NONSEGMENTED TABLESPACE IS INDICATING COMPRESSED, BUT YOUR SQL PA CONFIGURATION PARMS INDICATE THAT YOU DO NOT HAVE ESA HARDWARE COMPRESSION CAPABILITY ON THIS SYSTEM (ESACOMP NO). IF THIS IS THE PROPER PARM SETTING, SOFTWARE DATA COMPRESSION IS FAR MORE COSTLY.

User response: As appropriate.

*** ALERT: THIS PARTITIONED TABLESPACE IS NOT BEING ACCESSED USING ANY FORM OF PARALLELISM. POSSIBLE REASONS INCLUDE: INSUFFICIENT PARALLEL BUFFERS DEFINED IN POOL, OR DEGREE ANY NOT SPECIFIED ON THE BIND. ELAPSED TIME WOULD BE CONSIDERABLY ENHANCED BY CHOOSING A PARALLEL PROCESSING OPTION ON THIS PARTITIONED TABLESPACE.

User response: As appropriate.

*** ALERT: THE CURSOR WAS CONSIDERED AMBIGUOUS AT BIND TIME, MEANING THAT DB2 HAS NOT DECIDED HOW MANY DEGREES OF PARALLELISM WILL BE AVAILABLE AT RUN TIME. THE MAXIMUM NUMBER OF PARALLEL DEGREES IS EQUAL TO THE NUMBER OF PARTITIONS. PROGRAM MAY CHOOSE A CONSERVATIVE VALUE.

User response: As appropriate.

*** ALERT: THIS PARTITIONED TABLESPACE IS INDICATING COMPRESSED, BUT YOUR SQL PA CONFIGURATION PARMS INDICATE THAT YOU DO NOT HAVE ESA HARDWARE COMPRESSION CAPABILITY ON THIS SYSTEM (ESACOMP NO). IF THIS IS THE PROPER PARM SETTING, SOFTWARE DATA COMPRESSION IS FAR MORE COSTLY.

User response: As appropriate.

*** ALERT: THIS SEGMENTED TABLESPACE IS DESIGNATED AS COMPRESSED, BUT YOUR SQL PA CONFIGURATION PARMS INDICATE THAT YOU DO NOT HAVE ESA HARDWARE COMPRESSION CAPABILITY ON THIS SYSTEM (ESACOMP NO). IF THIS IS THE PROPER PARM SETTING, SOFTWARE DATA COMPRESSION IS FAR MORE COSTLY.

User response: As appropriate.
ANL5016W  *** WARNING: THIS TABLE IS
ACCESSSED BY ROWID DIRECTLY.
HOWEVER, IF THE PRIMARY ACCESS
PATH SHOULD FAIL, THE FALLBACK
ACCESS PLAN IS A TABLESPACE
SCAN. YOU SHOULD HAVE AN
INDEXED BACKUP PLAN IN CASE
THE ROWID IS CHANGED BY
REORG, ETC. TO SAFEGUARD
PERFORMANCE.

User response: As appropriate.

ANL5017W  *** ALERT: AT PRESENT, THE
PERCENTAGE OF ROWS ESTIMATED
to be read with this clustered
index via sequential prefetch
is approaching 50% of table
rows. Soon, DB2 will choose
tablespace scan instead of
matching the clustered index.
Runstats is recommended,
including collection of
non-uniform statistics. This
should culminate in better
stats, and allow DB2 to
continue to select this path.

User response: As appropriate.

ANL5018W  *** ALERT: THIS CLUSTERING
(insert) index has a low
cluster ratio (below 80)
and/or is not well clustered
(clustered flag = N), meaning
that all inserts try to follow
an unclustered pattern,
which is not helpful. You
should reorganize your table
and index, and re-run
runstats to update these
statistics. This should result
in better access paths and
performance in general.

User response: As appropriate.

ANL5019W  *** ALERT: PRESENTLY, THE
PERCENTAGE OF ROWS ESTIMATED
to be read with this random
index via list prefetch is
approaching 25% of total
table rows. Soon, DB2 will
choose tablespace scan
instead of matching the
random index. Runstats is
recommended, including the
collection of non-uniform
statistics. This should
culminate in better stats, and
allow DB2 to continue to
select this path.

User response: As appropriate.

ANL5020W  *** ALERT: A LOCKSIZE OF ROW HAS
BEEN EMPLOYED WITH A SELECT
STATEMENT. BE AWARE THAT THIS
CAUSES MORE LOCKING THAN
WITH LOCKSIZE PAGE, AND IF
THERE IS NOT A CONSIDERABLE
UPDATE REQUIREMENT, THIS
LOCKSIZE SHOULD NOT BE
EMPLOYED. EACH ROW LOCK
COSTS AS MUCH AS A PAGE LOCK,
AND COUNTS TOWARDS THE
MAXIMUM NUMBER OF LOCKS
HELD, ETC.

User response: As appropriate.

ANL5021W  *** WARNING: THIS IS A TYPE 1
INDEX STRUCTURE, WHICH HAS
MORE THAN 1 SUBPAGE DEFINED.
INSERTS TO THIS INDEX WILL
CAUSE LOGICAL PAGE SPLITS, AS
WELL AS PHYSICAL SPLITS.
CONVERSION TO TYPE 2 INDEX IS
SERIOUSLY RECOMMENDED, OR AT
LEAST REFORMAT USING SUBPAGES
OF 1 (4096 BYTES).

User response: As appropriate.

ANL5022W  *** ALERT: A LOCKSIZE OF
TABLESPACE IS NOT
RECOMMENDED WHEN YOU
EMPLOY DB2-ENFORCED
REFERENTIAL INTEGRITY.
CONSIDER CHANGING LOCKSIZE.

User response: As appropriate.
ANL5023W  *** ALERT: THIS MERGE SCAN JOIN IS NOT ELIGIBLE FOR EITHER CPU OR SYSPLEX PARALLELISM, SIMPLY BECAUSE IT JOINS ON MORE THAN ONE COLUMN.
User response: As appropriate.

ANL5024W  *** ALERT: THIS HYBRID JOIN IS NOT ELIGIBLE FOR MULTIPLE CPU OR SYSPLEX PARALLELISM SIMPLY BECAUSE IT SORTS THE INNER TABLE RID LIST.
User response: As appropriate.

ANL5025W  *** WARNING: ESTIMATE OF: cputim EXCEEDS CPU TIME LIMIT OF MAXCPU SECONDS!
User response: As appropriate.

ANL5026W  *** WARNING: ESTIMATE OF: qunits EXCEEDS SERVICE UNIT LIMIT OF maxqun QUNITS!
User response: As appropriate.

ANL5027W  *** WARNING: ESTIMATE OF: cost EXCEEDS MONETARY LIMIT OF maxcst money!
User response: As appropriate.

ANL5028W  *** WARNING: ESTIMATE OF: physio EXCEEDS I/O CALL LIMIT OF iollim I/O CALLS!
User response: As appropriate.

ANL5029W  *** WARNING: ESTIMATE OF: elapse EXCEEDS ELAPSED LIMIT OF maxelap SECONDS!
User response: As appropriate.

ANL5030W  *** WARNING: THE USE OF TRIGGERS ON THIS STATEMENT WILL CAUSE ADDITIONAL PROCESSING, BEYOND THE COST ESTIMATES CALCULATED: BE AWARE.
User response: As appropriate.

ANL5031W  *** WARNING: USER DEFINED FUNCTIONS (UDFs) ARE INVOKED DURING THIS STATEMENT EXECUTION, WHICH WILL ADD ADDITIONAL OVERHEAD TO THE FINAL COST.
User response: As appropriate.

ANL5032W  *** ALERT: LARGE OBJECTS ARE BEING REFERENCED IN THIS SQL STATEMENT FROM AN AUXILIARY TABLE. MATERIALIZING LOBS CAN BE A COSTLY PROCESS.
User response: As appropriate.

ANL5033W  *** ALERT: THIS IS A CORRELATED SUBQUERY WHICH IS INVOKING A TABLESPACE SCAN TO ACCESS DATA. CORRELATED SUBQUERIES ARE ALWAYS A STAGE 2 PREDICATE PROCESS. THEREFORE, CONSIDER REWRITING THIS SUBQUERY AS EITHER NON-CORRELATED OR AS A JOIN TO IMPROVE PERFORMANCE.
User response: As appropriate.

ANL5034W  *** ALERT: A MASS DELETE IS BEING PERFORMED AGAINST A NONSEGMENTED TABLESPACE. EACH ROW MUST BE INDIVIDUALLY DELETED AND LOGGED. UNLESS THE TABLESPACE PARTICIPATES IN RI, IT SHOULD BE CONVERTED TO A SEGMENTED TS TO ALLOW MASS DELETE TO CHANGE AND LOG ONLY THE SPACE MAP PAGES.
User response: As appropriate.

ANL5035W  *** WARNING: A DATA PARTITIONED SECONDARY INDEX (DPSI) IS BEING USED AS THE CLUSTERING INDEX FOR THIS TABLE. THIS COULD CAUSE ACCESS PATHS TO JUMP TO MULTIPLE PARTITIONS WHEN ACCESSING ROWS, LEADING TO AN EXPOSURE FOR SEQUENTIAL ACCESS. KEEP AN EYE ON NEAROFFPOSF AND FAROFFPOSF STATISTICS IN SYSINDEXPART.
User response: As appropriate.
ANL5036W  *** ALERT: THIS INDEX IS THE PARTITIONING INDEX FOR THE TABLE, BUT IT IS NOT DESIGNATED AS THE CLUSTERING INDEX. THIS MEANS THAT ANOTHER INDEX HAS BEEN CHOSEN FOR INSERT/LOAD ORDERING, WHICH COULD POSSIBLY WORSEN PERFORMANCE WHEN SEQUENTIALLY INSERTING DATA, BY JUMPING ACROSS MULTIPLE PARTITIONS.

User response: As appropriate.

ANL5037W  *** ALERT: REFRESH MQT TABLE PERFORMED ON A NONSEGMENTED TABLE SPACE. REFRESH INCLUDES A DELETE OF ALL ROWS – IF TABLE SPACE IS SEGMENTED THIS IS A MASS DELETE, JUST UPDATING THE SPACE MAP PAGES. HOWEVER, YOUR TABLE SPACE WILL CAUSE A LOG WRITE FOR EACH ROW DELETED – VERY COSTLY! CONVERT THE TABLE SPACE TO SEGMENTED FOR BETTER PERFORMANCE

User response: As appropriate.

ANL5038W  *** WARNING: THIS BASE TABLE HAS MQTS ASSOCIATED WITH IT, BUT THEY ARE NOT BEING CONSIDERED BY THE OPTIMIZER FOR AUTOMATIC QUERY REWRITE BECAUSE THE CURRENT REFRESH AGE IS SET TO 0. SET THE USER PARAMETER REFRESH ANY (ANLPARM) TO ENABLE MQT CONSIDERATION FOR THIS TABLE.

User response: As appropriate.

ANL5039W  *** WARNING: MULTIPLE DISTINCT SORTS HAVE BEEN DETECTED FOR THIS STATEMENT. ALTHOUGH V8+ ALLOWS LIBERAL USE OF THE DISTINCT KEYWORD, THIS SQL WILL ACTUALLY EXECUTE MULTIPLE SORTS, WHICH IS NOT DESIRABLE FOR GOOD PERFORMANCE.

User response: As appropriate.

ANL5040W  *** ALERT: THIS TABLE HAS BEEN DESIGNATED AS VOLATILE, SUGGESTING THAT DB2 SHOULD ALWAYS USE AN INDEX ON IT. HOWEVER, TABLE SPACE SCAN IS BEING SELECTED HERE. CHECK YOUR SQL CAREFULLY AND DETERMINE IF USING AN INDEXED COLUMN WILL IMPROVE YOUR ACCESS.

User response: As appropriate.

ANL5041W  *** ALERT: THIS MATERIALIZED QUERY TABLE (MQT) IS NOT BEING ACCESSED VIA AN INDEX SCAN, BECAUSE NO INDEXES EXIST ON THE MQT. BUILDING ONE OR MORE INDEXES ON THE MQT WILL RESULT IN BETTER PERFORMANCE.

User response: As appropriate.

ANL5042W  *** ALERT: SITE REQUIRES NOTIFICATION WHEN ANY TABLESPACE SCAN IS EMPLOYED.

User response: As appropriate.

ANL5043W  *** ALERT: SITE REQUIRES NOTIFICATION WHEN ANY NONMATCH IX SCAN IS EMPLOYED.

User response: As appropriate.

ANL5044W  *** ALERT: SITE REQUIRES NOTIFICATION WHEN ALL TS PARTITIONS ARE SCANNED.

User response: As appropriate.

ANL5045W  *** ALERT: THIS TABLE SPACE HAS THE NOT LOGGED ATTRIBUTE. UNDO AND REDO LOGGING FOR THE TABLE SPACE AND ALL INDEXES FOR TABLES IN THE TABLE SPACE IS SUPPRESSED. LOGGING IS ALSO SUPPRESSED FOR THE AUXILIARY INDEXES FOR ALL AUXILIARY TABLES ASSOCIATED WITH TABLES IN THE TABLE SPACE.

User response: As appropriate.

ANL5046W  *** Alert: Using the DECFLOAT CAST function results in a stage 2 predicate. Predicate transitive closure and indexing are not allowed.

User response: As appropriate.
Notes and recommendations

The following messages make general comments and recommendations on improving the performance of your queries. Notes and Recommendations are shown in reports if the ADVISOR parameter is set to YES or ALL in the user parameters file (ANLPARM).

These messages are self-explanatory; no additional explanation is given.

User response: As appropriate.

ANL5047W
*** Alert: Global optimization of this subquery has transformed the SQL statement and materialized a virtual table. Run time might be extended.
User response: As appropriate.

ANL5048W
*** Alert: XMLEXISTS is a stage 2 predicate unless an XML values index or a NODEID index is employed. Performance decreases when XML is used. The XMLEXISTS predicate is not a candidate for index screening.
User response: As appropriate.

ANL5049W
*** Alert: PRECISE NO will be deprecated and changed to PRECISE YES in DB2 V9 and later releases. When you use the PRECISE parameter, specify PRECISE YES or PRECISE ALL.
User response: As appropriate.

ANL5050W
*** Alert: DX access is not a candidate for any type of parallelism.
User response: As appropriate.

ANL5051W
*** Alert: XML is not allowed with SELECT DISTINCT, GROUP BY, or ORDER BY clauses, subselects that are not UNION ALL, quantified BETWEEN, DISTINCT, IN, or LIKE predicates, or aggregate DISTINCT functions.
User response: As appropriate.

Notes

ANL6001I
*** NOTE: PERFORMING UPDATES ON TABLES WHICH ARE HEAVILY INDEXED CAN HAVE AN IMPACT ON YOUR PERFORMANCE. WHEN TOO MANY INDEXES EXIST ON A TABLE AND THE COLUMNS SET ARE INDEXED, YOU CAN INCUR A LARGE OVERHEAD FOR INDEX MAINTENANCE. EACH INDEXED COLUMN Requires TWO CHANGES TO EACH AFFECTED INDEX: ONE TO DELETE THE OLD ENTRY, AND ANOTHER TO INSERT THE NEW VALUE. DO NOT OVER-INDEX FREQUENTLY UPDATED TABLES.
User response: As appropriate.

ANL6002I
*** NOTE: THIS QUERY HAS INTENTIONALLY TURNED OFF THE USE OF PREFETCH BY SPECIFYING A LOW VALUE ON THE OPTIMIZE FOR N ROWS CLAUSE. THIS PREVENTS SELECTION OF NOT ONLY SEQUENTIAL OR LIST PREFERENCES, BUT ALSO MULTI-INDEX ACCESS. IT ALSO DISCOURAGES BLOCK FETCH MODE FOR REMOTE CLIENT QUERIES. SEQUENTIAL DETECTION MAY INTERVENE LATER.
User response: As appropriate.

ANL6003I
*** NOTE: THIS QUERY HAS REQUESTED TO USE SELECTIVE PARTITION LOCKING BUT THE LOCK SIZE IS TABLESPACE, WHICH WILL NEGATE THIS FEATURE. SELECTIVE PARTITION LOCKING ALSO DOES NOT WORK WITH TYPE 1 INDEX STRUCTURES OR ACQUIRE (ALLOCATE) BIND PARAMETERS. FOR THIS QUERY TO USE SELECTIVE PARTITION LOCKS, SET LOCK SIZE TO PAGE OR TABLE.
User response: As appropriate.

ANL6004I
*** NOTE: THIS QUERY CONTAINS EITHER A MIN OR MAX BUILT-IN COLUMN FUNCTION AND HAS THE POTENTIAL TO CHOOSE AN II ACCESS METHOD, READING ONLY THE INDEX LEAF PAGE, BUT HAS NOT DONE SO. CONSIDER BUILDING INDEX THAT IS ASCENDING FOR MIN, OR DESCENDING FOR MAX, ON THE COLUMN(S) USED IN THE FUNCTION. ALL OTHER PRED PRED PRED PRED PRED MUST BE STAGE 1 AS WELL.
ANL6005I *** NOTE: THIS STATEMENT CONTAINS A BUILT-IN COLUMN FUNCTION; HOWEVER, IT IS NOT BEING PROCESSED AT EITHER STAGE 1 RETRIEVAL OR SORT TIME. IN GENERAL, THIS MEANS POOR PERFORMANCE, BECAUSE THE COLUMN(S) GET EVALUATED ONLY AT THE END OF STAGE 2 PROCESSING. THIS MIGHT BE DUE TO MULTIPLE COLUMNS USED, GROUP BY, NOT ALL STAGE 1 PREDs, ETC.

User response: As appropriate.

ANL6006I *** NOTE: THIS STATEMENT IS ACCESSING TABLE VIA A TABLESPACE SCAN, BUT NOT USING PREFETCH TO ACCESS THE PAGES. VERIFY THAT THE STATISTICS IN THE CATALOG ARE ACCURATE -- USUALLY A SMALL VALUE FOR NPAGES AND / OR NACTIVE WILL CAUSE THIS SITUATION.

User response: As appropriate.

ANL6007I *** RECOMMENDATION: HERE IS SOME ADVICE: CONVERT THE TABLESPACE TO A SEGMENTED DESIGN. THIS ALLOWS BETTER SPACE MANAGEMENT SCANNING ONLY ROWS THAT BELONG TO THE SINGLE TABLE REQUIRED AND ALSO PROMOTES CLUSTERING, ALLOWS LOCKSIZE TABLE, MASS DELETE, SPACE AVAILABILITY AFTER A DROP TABLE WITHOUT REORG, AND MUCH BETTER PERFORMANCE OVERALL.

User response: As appropriate.

ANL6008I *** RECOMMENDATION: IT IS POSSIBLE THAT THIS TABLESPACE SCAN MIGHT BE CAUSED BY THE ORING OF PREDICATES. WHEN OR DOES NOT CHOOSE MULTIPLE INDEX SCAN OR NON-MATCHING INDEX SCAN FOR STAGE 1 PREDICATES, IT MIGHT BE WORTH REWRITING THIS QUERY AS THE UNION OF SEVERAL STAGE 1 PREDs INSTEAD. CONSIDER THIS REWRITE TO AVOID THE TSCAN.

User response: As appropriate.

ANL6009I *** RECOMMENDATION: THIS TABLE APPARENTLY CONTAINS SOME VERY SMALL ROWS, BEING UNDER 24 BYTES IN SIZE, WHICH COULD ALLOW MORE THAN 127 ROWS TO BE STORED ON SINGLE PAGES: THIS COULD RESULT IN BETTER PERFORMANCE VIA LESS I/O. YOU MIGHT ALTER COMPRESS YES TO FORMAT PAGE FOR UP TO 255 ROWS, AND USE DATA COMPRESSION OR ALTER MAXROWS TO SPECIFY UP TO 255 ROWS PER PAGE. REORG THE TABLE, IN EITHER CASE, FOR CHANGES TO TAKE EFFECT.

User response: As appropriate.

ANL6010I *** NOTE: THIS QUERY COULD ONLY CHOOSE I/O PARALLELISM, DESPITE BEING IN A MORE PARALLEL ENVIRONMENT. POSSIBLE REASONS ARE: CORRELATED SUBQUERY, MERGE SCAN JOIN ON MORE THAN 1 COLUMN, UPDATEABLE CURSOR OR CURSOR WITH HOLD, HYBRID JOIN WITH SORTN_JOIN YES, UNION, LACK OF CPU ENGINES, OR ACCESS VIA TYPE 1 INDEX. REWRITE RECOMMENDED.

User response: As appropriate.

ANL6011I *** NOTE: THIS QUERY COULD ONLY CHOOSE SINGLE SYSTEM CP PARALLELISM, DESPITE BEING ELIGIBLE FOR SYSPLEX PARALLELISM AS MEMBER OF A DATA SHARING GROUP. POSSIBLE REASONS ARE: CORRELATED SUBQUERY, INDEX PROBES, IN LISTS, MERGE SCAN JOIN ON MORE THAN 1 COLUMN, UPDATEABLE CURSOR OR CURSOR WITH HOLD, HYBRID JOIN WITH SORTN_JOIN YES, UNION, RR OR RS ISOLATION, LIST PREFETCH WITH RANDOM OR MULTI-INDEX, EXISTS, OUTER JOINS, AMBIGUOUS CURSOR WITH CURRENTDATA YES, MATERIALIZED VIEWS OR GLOBAL TEMPORARY TABLES, OR SYSTEM IS NOT A DESIGNATED PARALLEL COORDINATOR. INVESTIGATE CAUSE AND REWRITE IF NEEDED.

User response: As appropriate.
*** RECOMMENDATION: BIG TABLESPACES DEFINED BY DSIZE OR LARGE SHOULD BE OPENED WITH CLOSE YES, AS EACH DATA AND INDEX PARTITION COUNT TOWARDS THE MAXIMUM NUMBER OF DATA SETS THAT CAN BE OPEN AT ONE TIME (10-32K THROUGH V7, OR 100K IN V8), WHICH MAY BE SUPERSEDED BY AN EVEN LOWER DSMAX PARM VALUE IN DB2.

User response: As appropriate.

*** RECOMMENDATION: THIS AMBIGUOUS CURSOR DOES NOT SEEM TO BE CAUSED BY HOST VARIABLES. THEREFORE, IT IS LIKELY THAT YOU CAN IMPROVE YOUR PERFORMANCE AND ELIMINATE AMBIGUITY BY CODING FOR FETCH ONLY ON THIS SELECT STMT.

User response: As appropriate.

*** NOTE: MASS DELETES (ALL ROWS) FROM SEGMENTED TABLESPACES CAN BE NEGATED BY USE OF REFERENTIAL INTEGRITY, VALIDATION PROCS, OR THE REMOTE RECOVERY DATA FACILITY (RRDF) DATA CAPTURE CHANGES OPTION...MAKE SURE THAT YOU DO NOT FALL INTO ONE OF THESE EXCEPTION CATEGORIES.

User response: As appropriate.

*** NOTE: THE SEGMENTED TABLE IS MORE THAN 32 PAGES IN SIZE, YET THE SEGSIZE IS BELOW 32 PAGES. IN A MULTITABLE TABLESPACE, THE FINAL SEQUENTIAL PREFETCH COULD INADVERTENTLY READ PAGES IN THE NEXT TABLES SEGMENT BUT YOU CAN AVOID THIS SITUATION BY USING A SEGSIZE OF 32 OR 64.

User response: As appropriate.

*** NOTE: PRESENTLY, THIS QUERY IS ONLY MATCHING SOME OF THE Indexed columns available with this index. Maximizing the use of predicates against these index columns will improve performance, either by matching or screening rows and therefore reducing data page I/O requirements.

User response: As appropriate.

*** NOTE: THIS QUERY IS CURRENTLY PERFORMING A NONMATCHING INDEX SCAN, WHICH CAUSES IT TO READ EVERY INDEX LEAF PAGE TO PROCESS THE PREDICATES. READING THE ENTIRE LEAF TREE IS TIME-CONSUMING VS. MATCHING A PART OF THE INDEX. CONSIDER BETTER PREDICATES THAT ALLOW YOU TO MATCH.

User response: As appropriate.

*** RECOMMENDATION: THIS NONMATCHING INDEX SCAN ACCESS PATH MIGHT BE CAUSED BY THE ORING OF PREDICATES. WHEN OR DOES NOT CHOOSE MULTIPLE INDEX SCAN OR NON-MATCHING INDEX SCAN FOR STAGE 1 PREDICATES, IT MIGHT BE WORTH REWRITING THIS QUERY AS THE UNION OF SEVERAL STAGE 1 PREDS INSTEAD. CONSIDER THIS REWRITE TO AVOID THE TSCAN.

User response: As appropriate.

*** RECOMMENDATION: OLD INDEX TYPE FOUND: TYPE 1 INDEXES SHOULD BE CONVERTED TO TYPE 2 TO ELIMINATE INDEX LOCKING ON INSERTS, UPDATES AND DELETES. THIS INDEX SHOULD BE CONVERTED TO TYPE 2 AS SOON AS POSSIBLE.

User response: As appropriate.
**ANL6020I** *** NOTE: THIS INDEX IS PRESENTLY DESIGNATED AS ALLOWING DUPLICATE VALUES. MAKE SURE THIS IS A NONUNIQUE INDEX THAT DOES NOT FALL INTO ONE OF THE FOLLOWING UNIQUE CATEGORIES: EXPLICITLY UNIQUE, PRIMARY KEY, NONPRIMARY RI PARENT KEY, UNIQUE WHERE NOT NULL, UNIQUE COLUMN CONSTRAINT. UNIQUE INDEXES HAVE DEFINITE ADVANTAGES WHERE EVER THEY CAN BE DECLARED BY THE USER, SO BE SURE IT CONTAINS DUPLICATES.

**User response:** As appropriate.

---

**ANL6021I** *** NOTE: THIS TABLESPACE IS DESIGNATED COMPRESSED, BUT YOUR INDEX ACCESS IS RANDOM...THE LIKELIHOOD OF FINDING MANY ROWS PER PAGE TO SATISFY YOUR QUERY IS REMOTE. THUS, DATA COMPRESSION IS NOT REDUCING YOUR I/O SIGNIFICANTLY (OK FOR STORAGE REDUCTION). PREFER CLUSTER INDEX.

**User response:** As appropriate.

---

**ANL6022I** *** RECOMMENDATION: ANY TYPE 1 NON-PARTITIONED SECONDARY INDEXES BUILT UPON PARTITIONED TABLESPACES DO NOT ALLOW CONCURRENT ACCESS, AND MAY DEFEAT PARALLEL EFFORTS. CONVERT THIS INDEX TO TYPE 2, TO TAKE ADVANTAGE OF LOGICAL PARTITIONING AVAILABLE VIA THE EXTRA RID BYTE, WHICH HOLDS PART NO.

**User response:** As appropriate.

---

**ANL6023I** *** RECOMMENDATION: YOUR LARGER NON-PARTITIONED SECONDARY INDEXES BUILT ON PARTITIONED TABLESPACES CAN BE PHYSICALLY BROKEN UP AND DISTRIBUTED ACROSS MANY DISK DRIVES USING THE PIECESIZE PARAMETER TO DESIGNATE THE SIZE OF EACH SLICE OF THE INDEX TO BE STORED ON A DIFFERENT DATA SET.

**User response:** As appropriate.

---

**ANL6024I** *** NOTE: STAGE 2 PREDICATE(S) HAVE BEEN DETECTED IN STATEMENT--THESE DO NOT PERFORM AS WELL AS STAGE 1 AND CONSUME EXTRA CPU PROCESSING.

**User response:** As appropriate.

---

**ANL6025I** *** NOTE: THE COMPOSITE (FINAL) FILTER FACTOR INDICATES THAT A LARGE NUMBER OF ROWS WILL BE RETURNED.

**User response:** As appropriate.

---

**ANL6026I** *** NOTE: ARITHMETIC EXPRESSIONS THAT ARE NOT EXCLUSIVELY USING INTEGERS WILL BE STAGE 2 AND NOT MATCHING INDEXABLE. MOVE TO SELECT LIST IF POSSIBLE, OR DO THE MATH IN YOUR APPLICATION LATER.

**User response:** As appropriate.

---

**ANL6027I** *** NOTE: THE REFERENTIAL DELETE RULE OF CASCADE ON THE PARENT TABLE WILL FORCE CASCADED DELETES TO THIS DEPENDENT TABLE AS WELL.

**User response:** As appropriate.

---

**ANL6028I** *** NOTE: THE REFERENTIAL DELETE RULE OF SET NULL ON THE PARENT TABLE WILL FORCE CASCADED UPDATES TO THIS DEPENDENT TABLE AS WELL.

**User response:** As appropriate.

---

**ANL6029I** *** NOTE: THE REFERENTIAL DELETE RULE OF RESTRICT WILL NOT ALLOW ROWS TO BE DELETED FROM THIS TABLE IF ANY DEPENDENT ROWS EXIST.

**User response:** As appropriate.

---

**ANL6030I** *** NOTE: THE REFERENTIAL DELETE RULE OF NO ACTION WILL ONLY ALLOW SELF-REFERENCING DELETES TO THIS TABLE, BUT NOT TO DEPENDENTS.

**User response:** As appropriate.
ANL6031I  *** NOTE: THE REFERENTIAL DELETE RULE OF CASCADE WILL FORCE CORRESPONDING ROWS TO BE DELETED FROM THIS TABLE IF ANY DEPENDENT ROWS EXIST.

User response: As appropriate.

ANL6032I  *** NOTE: THE REFERENTIAL DELETE RULE OF SET NULL WILL FORCE CORRESPONDING ROWS TO BE UPDATED IN THIS TABLE IF ANY DEPENDENT ROWS EXIST.

User response: As appropriate.

ANL6033I  *** NOTE: EITHER THE NESTED LOOP OR MERGE SCAN JOIN METHODS ARE THE ONLY POSSIBILITIES WHEN LEFT OR RIGHT OUTER JOIN HAS BEEN REQUESTED.

User response: As appropriate.

ANL6034I  *** NOTE: ONLY THE NESTED LOOP JOIN METHOD SUPPORTS THE USE OF NON-EQUIJOIN PREDICATES (T1.C1 <= T2.C2). IF YOU ARE USING THESE PREDICATES TO JOIN TABLES TOGETHER, YOU ARE LIMITING DB2 TO SELECTING ONLY NL...MERGE SCAN AND HYBRID REQUIRE EQUIJOIN PREDs (T1.C1 = T2.C2).

User response: As appropriate.

ANL6035I  *** NOTE: THE MERGE SCAN JOIN METHOD IS ALWAYS EMPLOYED WHENEVER A FULL OUTER JOIN HAS BEEN REQUESTED.

User response: As appropriate.

ANL6036I  *** NOTE: THIS IS A SUBQUERY PROCESS APPLIED TO A PARTITIONED TABLE. NO PARALLEL OPERATIONS ARE POSSIBLE BECAUSE IT INCLUDES THE EXISTS PREDICATE, WHICH NEGATES THE USE OF PARALLELISM.

User response: As appropriate.

ANL6037I  *** NOTE: THIS IS A NONCORRELATED SUBQUERY WHICH IS USING THE IN PREDICATE. NO PARALLEL OPERATIONS WERE CHOSEN ON THIS PARTITIONED TABLESPACE: THEREFORE, IT IS POSSIBLE THAT DB2 IS BUILDING A SPARSE INDEX ON THEN INNER QUERY COLUMN, TO SPEED UP ACCESS TO THE INNER TABLE.

User response: As appropriate.

ANL6038I  *** NOTE: THIS IS A CORRELATED SUBQUERY PROCESSED UPON A PARTITIONED TABLE. NO PARALLEL OPERATIONS ARE POSSIBLE WITH CORRELATED SUBQUERIES. THEREFORE, CONSIDER REWRITING AS NON-CORRELATED, OR AS A JOIN, SO THAT YOU CAN TAKE ADVANTAGE OF PARALLEL OPERATIONS.

User response: As appropriate.

ANL6039I  *** NOTE: SCROLLABLE CURSORS DO NOT SUPPORT PARALLELISM AND CANNOT BE ACCESSED VIA PRIVATE PROTOCOL (DB2 MVS-MVS). IN ADDITION, BE AWARE THAT BUILT-IN FUNCTIONS DO NOT RE-EVALUATE (AVG STAYS CONSTANT) BUT ARITHMETIC EXPRESSIONS WILL BE RE-EVALUATED FOR EACH ROW FETCHED. DB2 USES DECLARED TEMP TABLES FOR PROCESSING STATIC SCROLLABLE CURSORS.

User response: As appropriate.

ANL6040I  *** NOTE: THIS STATEMENT WAS DECLARED WITH AN INSENSITIVE SCROLLABLE CURSOR. THE SIZE, THE ORDER OF ROWS AND THE VALUES FOR EACH ROW OF RESULT TABLE DO NOT CHANGE AFTER THE CURSOR IS OPENED. THE RESULT TABLE IS READ ONLY, AND THE CURSOR CANNOT BE USED FOR UPDATE | DELETE.

User response: As appropriate.
ANL6041I  *** NOTE: THIS STATEMENT WAS DECLARED WITH A SENSITIVE STATIC SCROLLABLE CURSOR. POSITIONED UPDATES AND DELETES WILL BE VISIBLE IN THE RESULT TABLE AND CHANGES MADE TO THE UNDERLYING TABLE BY OTHER CURSORS MAY OR MAY NOT BE VISIBLE, DEPENDING UPON WHETHER FETCH IS SENSITIVE OR INSENSITIVE TO CHANGES. INSERT TO UNDERLYING TABLE IS NOT VISIBLE.

User response:  As appropriate.

ANL6042I  *** NOTE: DYNAMIC STATEMENT CACHING IS ASSUMED FOR THIS SET OF ESTIMATES.

User response:  As appropriate.

ANL6043I  *** NOTE: IMPROVEMENTS TO OUTER JOIN PROCESSING IN THIS AND SUBSEQUENT DB2 RELEASES INCLUDE ABILITY TO USE PREDICATE TRANSITIVE CLOSURE AND APPLY PREDICATES FROM THE WHERE CLAUSE, PLUS OTHER STREAMLINING.

User response:  As appropriate.

ANL6044I  *** NOTE: CONNECTION VIA DRDA IS INDICATED, AND OPTIMIZE FOR N ROWS IS ALSO DEPLOYED. A MINIMUM NUMBER OF ROWS MAY BE RETURNED (16) PER FETCH.

User response:  As appropriate.

ANL6045I  *** NOTE: THIS INDEX IS THE PRIMARY KEY: YOU MUST DROP EITHER THE KEY OR THE REFERENTIAL CONSTRAINT BEFORE DROPPING THIS INDEX.

User response:  As appropriate.

ANL6046I  *** NOTE: IN (LIST) PREDICATE IS BEING PROCESSED IN PARALLEL, RESULTING IN IMPROVED ELAPSED TIME AND PERFORMANCE.

User response:  As appropriate.

ANL6047I  *** NOTE: THIS NC SUBQUERY IS BEING PROCESSED USING AN INDEX ON THE OUTER COLUMN. FIRST, INNER QUERY IS EXECUTED AND SORTED INTO THE OUTER COLUMN ORDER. THEN, FOR EACH VALUE IN THE INNER TABLE, THE OUTER INDEX IS SCANNED FOR MATCHES, REVERSING USUAL NONCORRELATED ORDER.

User response:  As appropriate.

ANL6051I  *** NOTE: ONCE A STAR JOIN QUALIFIES, NO OTHER JOIN METHOD IS CONSIDERED. QUALIFICATIONS INCLUDE A SIGNIFICANTLY LARGER FACT TABLE (25X DEFAULT), AT LEAST DIMENSION TABLES, ALL EQUI-JOIN BOOLEAN TERM PREDICATES BETWEEN THE FACT AND DIMENSIONS, NO CORRELATED SUBQUERY CROSSES DIMENSIONS, NO OUTER JOINS ARE LEFT AFTER QUERY REWRITE, AND DATA TYPE AND LENGTHS ARE THE SAME FOR ALL THE JOIN PREDICATES.

User response:  As appropriate.

ANL6052I  *** NOTE: THIS INDEX IS SPECIFIED AS NOT PADDED, ALLOWING STORAGE OF A VARYING LENGTH INDEX KEY. PADDED INDEXES USE BLANKS TO FILL OUT THEIR FIXED LENGTH KEYS AND ARE NOT ELIGIBLE FOR INDEX ONLY. NOT PADDED INDEXES DO NOT BLANK FILL CHAR/VARCHAR COLUMNS ALLOWING GREATER FLEXIBILITY AND BETTER USE OF STORAGE, PACKING MORE ENTRIES INTO A SINGLE LEAF PAGE. INDEX ONLY ACCESS IS PERMITTED ON NOT PADDED INDEXES.

User response:  As appropriate.
**ANL6054I**  **ANL6051I**

User response: As appropriate.

**ANL6054I**  *** NOTE: THIS INDEX IS DEFINED AS THE PRIMARY PARTITIONING INDEX FOR THE TABLE. THE TRADITIONAL INDEX-BASED PARTITIONING SCHEME IS CONTROLLED BY THIS INDEX.

User response: As appropriate.

**ANL6055I**  *** RECOMMENDATION: THIS MQT IS NOT CREATED IN A SEGMENTED TABLE SPACE. ATTEMPTS TO REFRESH THIS TABLE WILL CAUSE LARGE OVERHEAD FROM A MASS DELETE OF ALL ROWS PRIOR TO REPOPULATION. CONVERT TO A SEGMENTED TABLE SPACE

User response: As appropriate.

**ANL6056I**  *** NOTE: THIS STATEMENT WAS DECLARED WITH A DYNAMIC SCROLLABLE CURSOR, WHICH ALLOWS SCROLLING DIRECTLY ON THE BASE TABLE AND IS SENSITIVE TO ALL COMMITTED INSERTS, UPDATES AND DELETES. USERS DO NOT HAVE TO COMMIT THEIR CHANGES TO SEE THEM WITH DYNAMIC SCROLLING

User response: As appropriate.

**ANL6057I**  *** NOTE: INDEXED ACCESS ON A SCROLLABLE CURSOR CAN AVOID ORDER BY SORT WHEN THE KEY IS AN EXACT MATCH OR EXACT OPPOSITE (FOR BACKWARD SCAN). THUS INDEX ON C1 ASC, C2 DESC WILL AVOID ORDER BY C1 ASC, C2 DESC FORWARD SCAN, AND EMPLOY BACKWARD SCAN TO AVOID ORDER BY C1 DESC, C2 ASC. HOWEVER, SORTS MUST BE PERFORMED FOR ORDER BY C1 DESC, C2 ASC OR ORDER BY C1 ASC, C2 ASC AS THEY ARE NOT EXACT MATCHES OR OPPOSITES.

User response: As appropriate.

**ANL6058I**  *** RECOMMENDATION: OPPOSITE INDEXES BUILT WITH COLUMNS IN REVERSE ORDER CAN NOW BE DELETED AS DB2 DOES NOT REQUIRE THEM FOR ORDER BY OR SEQUENTIAL SCANS: INDEXES CAN NOW BE SCANNED BACKWARDS. THIS RECOMMENDATION ONLY APPLIES TO TRUE OPPOSITE INDEXES THAT INCLUDES REVERSAL OF ASC AND DESC ORDER.

User response: As appropriate.

**ANL6059I**  *** RECOMMENDATION: CHANGE THE SEGSIZE OF THIS TABLE SPACE TO 32 PAGES. MOST TABLES BETWEEN 28 AND 127 PAGES WILL BENEFIT FROM THIS CHANGE BY MAXIMIZING PREFETCH PERFORMANCE. DO THIS UNLESS TABLE SPACE CONTAINS MANY OTHER SMALLER TABLES.

User response: As appropriate.

**ANL6060I**  *** RECOMMENDATION: CHANGE THE SEGSIZE OF THIS TABLE SPACE TO 64 PAGES. ANY TABLE THAT IS AT LEAST 128 PAGES WILL BENEFIT FROM THIS CHANGE BY MAXIMIZING PREFETCH PERFORMANCE. DO THIS UNLESS TABLE SPACE CONTAINS MANY OTHER TABLES < 128 PAGES IN SIZE.

User response: As appropriate.

**ANL6061I**  *** RECOMMENDATION: A DPSI INDEX IS USED TO ACCESS TABLE, BUT LIMITED PAGE RANGE SCAN WAS NOT PERMITTED. TO IMPROVE CHANCES OF PRUNING SOME PARTITIONS FROM SCAN SET, TRY REOPT(VARS) OR REOPT(ONCE) TO FORCE REEVALUATION OF HOST VARIABLES.

User response: As appropriate.
**Chapter 11. Messages and return codes**

---

**ANL6062I**  *** NOTE: PARALLEL DATA ACCESS IS BEING PERFORMED, BUT PARALLEL SORTING WAS NOT SELECTED. TO ENABLE PARALLEL SORTING, THE TOTAL SORT DATA SIZE MUST BE > 2 MB (OR 500 PAGES), AND/OR THE SORT DATA PER PARALLEL DEGREE MUST BE > 100KB (OR 25 PAGES). IF YOUR DATA EXCEEDS THESE SIZES THEN ENSURE THAT THE ZPARM QWP40PSE IS ENABLED.

User response: As appropriate.

**ANL6063I**  *** NOTE: THIS STATEMENT CONTAINS A MULTI-ROW FETCH, IN WHICH ROWS ARE GROUPED IN A SET AND OPERATIONS CAN BE PERFORMED ON THE CURRENT ROW SET, OR THE USER CAN FETCH THE NEXT SET.

User response: As appropriate.

**ANL6064I**  *** NOTE: A MULTI-ROW INSERT HAS BEEN DETECTED WITH THE ATOMIC ATTRIBUTE: IN THE EVENT OF ANY FAILURE, ALL CHANGES WILL BE ROLLED BACK.

User response: As appropriate.

**ANL6065I**  *** NOTE: A MULTI-ROW INSERT HAS BEEN DETECTED WITHOUT THE ATOMIC ATTRIBUTE: ANY INSERTED ROWS WILL BE COMMITTED, EVEN IF SOME INSERTS FAIL.

User response: As appropriate.

**ANL6066I**  *** NOTE: A SIGNIFICANT NUMBER OF DATA PAGES (SITE DEPENDENT) ARE READ WITH THIS TABLESPACE SCAN OPERATION.

User response: As appropriate.

**ANL6067I**  *** NOTE: A SIGNIFICANT NUMBER OF INDEX LEAF PAGES (SITE DEPENDENT) ARE READ WITH THIS MATCHING INDEX SCAN OPERATION.

User response: As appropriate.

**ANL6068I**  *** NOTE: A SIGNIFICANT NUMBER OF INDEX LEAF PAGES (SITE DEPENDENT) ARE READ WITH THIS NON-MATCHING INDEX SCAN OPERATION.

User response: As appropriate.

**ANL6069I**  *** NOTE: BUILDING AN INDEX WITH THE SAME ORDERING OF SORT KEY COLUMNS WILL NEGATE THE SORT REQUESTED HERE...CONSIDER AS WARRANTED.

User response: As appropriate.

**ANL6070I**  *** NOTE: THE TABLE APPEND FLAG IS SET. THEREFORE ALL INSERTS WILL BE DONE AT THE END OF THIS TABLE SPACE.

User response: As appropriate.

**ANL6071I**  *** NOTE: THIS IS A PARTITION-BY-GROWTH, WHICH HAS THE BENEFITS OF ADDING PARTITIONS AS REQUIRED, PLUS THE INTERNAL FORMAT OF A SEGMENTED TABLESPACE.

User response: As appropriate.

**ANL6072I**  *** NOTE: THIS IS A UNIVERSAL TABLE SPACE (UTS), WHICH HAS THE BENEFITS OF USING THE INTERNAL FORMAT OF A SEGMENTED TABLE SPACE (MASS DELETES ON SPACE MAP ONLY, BETTER SPACE MANAGEMENT, ETC.).

User response: As appropriate.

**ANL6073I**  *** NOTE: REFERENTIAL INTEGRITY PRIMARY AND FOREIGN KEYS CANNOT BE CREATED AS "INDEX ON EXPRESSION" TYPE.

User response: As appropriate.

**ANL6074I**  *** NOTE: INDEX ON EXPRESSION WAS USED AS THE ACCESS PATH FOR THIS QUERY BLOCK.

User response: As appropriate.
Good news and guidelines

The following messages are intended to guide newer programmers toward good query coding. **Good News and Guidelines** are active if the ADVISOR parameter is set to ALL in the user parameters file (ANLPARM).

The messages are self-explanatory; no additional explanations are shown.

**Guideline**: Make sure that your frequently updated columns are being kept together in the table row: DB2 logs from the first to last changed byte. Also, put VARCHAR at the end of the row: DB2 logs from the first changed byte to end of row. Length changes cause the entire row to log, as length field is kept with the row header. Rows that are compressed are variable length & logged in compressed format.

**Guideline**: Close YES was specified for the tablespace and/or the index...If these are little used this is ok. If high volume access then consider close NO. Extremely relevant pages can be page fixed in memory by highly referencing, using Hiperpools (through V7), putting into a dedicated buffer pool large enough to hold all pages, deploying data sharing with group buffer pool cache all option, etc. Each with associated costs. Close NO also increases chances that DBD will remain in EDM pool for next execution.

**Guideline**: Resetting counters at commit.

**Guideline**: This plan step has not selected any sequential | list prefetch I/O. If the SQL processes just a few rows that is ok, but if many rows are involved, you can help promote sequential detection by both accessing the data in sequential order (PRESORT?) and by binding with release (DEALLOCATE) to avoid resetting counters at commit.
ANL7004I  *** GOOD NEWS: THIS QUERY IS ACCESSING A TABLE AND/OR INDEX IN PARALLEL EMPLOYING A LIMITED PAGE RANGE SCAN. THIS MEANS THAT NOT ALL PARTITIONS ARE NECESSARY TO RESOLVE THE QUERY. INSTEAD, DB2 WILL FIND THE ANSWER SET IN ONLY A SUBSET OF THE PARTITIONS OF THE TABLE AND/OR INDEX.

User response: As appropriate.

ANL7005I  *** GOOD NEWS: THIS SQL FEATURES A UNIQUE INDEX BEING USED TO ACCESS THE TABLE THAT CAN AVOID DISTINCT SORTS WITH GROUP BY AND COUNT DISTINCT C1. ALSO, UNIQUE INDEX WILL AVOID 4K IN MEMORY TABLE ON CORRELATED SUBQUERY, CONSIDERS SPECIAL ACCESS PATH WHEN USED WITH = PRED, AND MAY CONVERT SUBQUERY TO JOIN ON IN, =ANY OR =SOME PREDICATES.

User response: As appropriate.

ANL7007I  *** This statement contains a select of all columns in the table. Typically a select of all columns results in significantly increased processing time to fetch and process each column of each row. Use select all only when you want to select all columns in a view definition. Improve performance by specifying the name of each column for the SQL SELECT statement.

Explanation: Using the SELECT ALL statement results in slower query performance when compared to queries that specify the column names.

User response: Modify your queries to specify the column names in the SELECT statements.

ANL7008I  *** GUIDELINE: ATTEMPT TO PARTITION YOUR TABLESPACES EVENLY, AND MAKE ENOUGH PARTITIONS TO EXPLOIT THE MAXIMUM NUMBER OF PROCESSORS ONLINE.

User response: As appropriate.

ANL7009I  *** GUIDELINE: ON COMPOSITE (MULTI-COLUMN) INDEXES, FAVOR MATCHING INDEX SCAN OF AS MANY COLUMNS AS POSSIBLE BY USING EQUALS (=), RANGE (>, <, >=, <=, BETWEEN, LIKE), IN (LIST) AND NULL PREDICATES, WHERE A RANGE PRED IS THE LAST MATCHING PREDICATE. APPLY LEFT TO RIGHT AGAINST INDEX COLUMNS, SKIPPING NO COLUMNS IN THE L-R SEQUENCE. DB2 MAY SCREEN REMAINING STAGE 1 PREDS AGAINST THE RID LIST BEFORE DATA ACCESS.

User response: As appropriate.

ANL7010I  *** GUIDELINE: ON COMPOSITE (MULTI-COLUMN) INDEXES, COLLECT CORRELATED KEY STATS ON COLUMN PAIRINGS, AS WELL AS CARDINALITY STATS LIKE 2NDKEYCARD, 3RDKKEYCARD, ETC. WITH RUNSTATS. THESE ADDITIONAL STATISTICS WILL ALLOW DB2 TO MAKE BETTER FILTER ESTIMATES FOR EQUALITY AND RANGE AND IN LIST PREDICATES, THEREBY SELECTING BETTER INDEXES AND MORE REALISTIC ESTIMATES FOR ROWS RETURNED FROM THE QUERY (QCARD).

User response: As appropriate.

ANL7011I  *** GUIDELINE: THIS OLD TYPE 1 INDEX FORMAT SHOULD BE CONVERTED INTO TYPE 2 TO ELIMINATE INDEX LOCKING. TYPE 1 INDEXES DO NOT ALLOW: LOCKING AT THE ROW LEVEL, UR ISOLATION, CP OR SYSPLEX PARALLELISM, LOGICAL PARTITIONING OF NONPARTITIONED SECONDARY INDEXES, ETC.

User response: As appropriate.
ANL7012I *** GUIDELINE: MAKE SURE THAT YOU HAVE DEFINED ADEQUATE FREE SPACE PARMS IN THIS INDEX STRUCTURE, AS WELL AS IN THE DATA TABLE, TO KEEP ROWS IN THE CLUSTERED ORDER AND AVOID FRAGMENTATION. BOTH FREEPAGE AND PCTFREE SHOULD BE DEPLOYED, AND RUNSTATS PERIODICALLY RUN TO UPDATE STATS.

User response: As appropriate.

ANL7013I *** GUIDELINE: DO NOT USE ABSOLUTE CURSOR POSITIONING WHEN YOU ARE UPDATING ONLY A SINGLE ROW, UNLESS YOU HAVE SELECTED THE ROW TO GUARANTEE THAT NO CHANGES WILL OCCUR BETWEEN THE SELECT AND UPDATE. ELSE, IT IS MORE TIME CONSUMING FOR DB2 TO POSITION THE CURSOR RATHER THAN TO DO A DIRECT UPDATE ON THE ROW. ALSO, CONSIDER KEEP UPDATE LOCKS AS A MEANS OF SERIALIZING UPDATES AND COMPLETING THEM QUICKLY.

User response: As appropriate.

ANL7014I *** GUIDELINE: DO NOT USE ABSOLUTE CURSOR POSITIONING WHEN YOU ARE DELETING ONLY A SINGLE ROW, UNLESS YOU HAVE SELECTED THE ROW TO GUARANTEE THAT NO CHANGES WILL OCCUR BETWEEN THE SELECT AND DELETE. ELSE, IT IS MORE TIME CONSUMING FOR DB2 TO POSITION THE CURSOR RATHER THAN TO DO A DIRECT DELETE ON THE ROW.

User response: As appropriate.

ANL7015I *** GUIDELINE: COLLECTING NON-UNIFORM COLUMN STATISTICS GREATLY ENHANCE THE DB2 OPTIMIZER ACCURACY WHEN PROCESSING EQUALS AND IN (LIST) PREDs.

User response: As appropriate.

ANL7016I *** GUIDELINE: CONSIDER BIND OPTION REOPT (VARS) REOPT(ONCE) IN V8+ TO HELP ENHANCE PERFORMANCE OF POORLY RUNNING SQL USING HOST VARIABLES, REOPTIMIZING AT RUNTIME.

User response: As appropriate.

ANL7017I *** GUIDELINE: THIS PROCESS JOINS A SIGNIFICANT NUMBER OF TABLES TOGETHER: THE EXPOSURE TO PERFORMANCE PROBLEMS IS INCREASED. IN GENERAL, YOU SHOULD TRY TO KEEP TO A REASONABLE NUMBER FOR OPTIMUM QUERY PERFORMANCE. LIMIT IS 15 TABLES FOR NORMAL JOIN, (UP TO 225 in V8+), 225 FOR STAR JOIN.

User response: As appropriate.

ANL7018I *** GUIDELINE: IF YOU WISH TO REVERSE THE NESTED LOOP JOIN TABLE ORDER, USE SOME REDUNDANT PREDICATES (TN.CN = TN.CN) ON THE TABLE YOU WOULD LIKE TO BECOME THE OUTER TABLE IN THE PROCESS (REVERSE THIS LOGIC FOR MERGE SCAN JOINS, APPLYING THEM TO THE INNER TABLE INSTEAD).

User response: As appropriate.

ANL7019I *** GUIDELINE: TO SELECT NESTED LOOP OR HYBRID JOIN INSTEAD OF THE MERGE SCAN, YOU CAN TRY THE OPTIMIZE FOR N ROWS CLAUSE TO MAKE THE DB2 OPTIMIZER THINK THAT THESE TABLES ARE SMALLER, AND THE RESULTING ANSWER SET IS ALSO SMALL. MS IS LIKELY SELECTED WHEN MORE ROWS WILL RESULT, OR WHEN NO INDEXES ARE PRESENT OR USABLE.

User response: As appropriate.

ANL7020I *** GOOD NEWS: THIS SUBQUERY HAS BEEN AUTOMATICALLY TRANSFORMED BY DB2 INTO A SIMPLE PREDICATE, ELIMINATING THE SUBQUERY PROCESSING OVERHEAD.

User response: As appropriate.
| ANL7021I | *** GOOD NEWS: THIS CORRELATED SUBQUERY HAS A UNIQUE INDEX ON ONE|BOTH TABLES AND THEREFORE WILL NOT HAVE TO BUILD A 4K IN MEMORY TABLE TO STORE RESULTS OF PREVIOUS CORRELATIONS. BEING UNIQUE, THE CHANCES OF FINDING A MATCH ARE NIL, SO DB2 AVOIDS THE UNNECESSARY OVERHEAD. |
| --- | --- |
| User response: | As appropriate. |

<table>
<thead>
<tr>
<th>ANL7022I</th>
<th>*** GUIDELINE: AVOID REFERENCING VARCHAR COLUMNS IN THE SORT, AS THESE WILL BE PADDED TO MAXIMUM SIZE IN THE SORT WORK RECORDS. LONG SORT RECORDS OVER 4000 BYTES CAUSE A TAG SORT, WHICH REQUIRES EVEN MORE OVERHEAD TO PARK RECORDS IN A FILE AND READ THEM AFTERWARD.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User response:</td>
<td>As appropriate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL7023I</th>
<th>*** GOOD NEWS: EQUI-JOIN PREDICATES OF UNEQUAL LENGTH MAY BE HANDLED AT STAGE 1: DB2 PADS OUT CHAR AND VARCHAR COLUMN TYPES TO ACCOMPLISH THE JOIN.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User response:</td>
<td>As appropriate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL7024I</th>
<th>*** GOOD NEWS: INDEX SCREENING IS AVAILABLE FOR LIST PREFETCH PROCESSING, WHICH ALLOWS RANDOM INDEXES TO SCREEN ADDITIONAL PREDICATES AFTER MATCHING INDEX SCAN.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User response:</td>
<td>As appropriate.</td>
</tr>
</tbody>
</table>

| ANL7025I | *** GOOD NEWS: ORDER BY SORTS NO LONGER NEED TO SPECIFY SORT KEY COLUMNS IN THE SELECT LIST, BEGINNING IN V5R2+. HOWEVER, THE RESTRICTION STILL APPLIES WITH ORDER BY USED IN CONJUNCTION WITH UNIONS, OR GROUP BY|DISTINCT SORTS AND MIN, MAX OR SUM BUILT-IN FUNCTIONS. |
| --- | --- |
| User response: | As appropriate. |

<table>
<thead>
<tr>
<th>ANL7026I</th>
<th>*** GUIDELINE: THIS IS A MATERIALIZED QUERY TABLE (MQT) WHICH CONTAINS SUMMARY DATA. MQTS CAN BE USER OR SYSTEM MAINTAINED – BOTH METHODS CAN USE THE REFRESH TABLE STATEMENT, BUT USER-MAINTAINED ALSO ALLOW INSERT, UPDATE, DELETE AND LOAD UTILITY. CURRENT REFRESH AGE ANY MUST BE SET TO ALLOW CONSIDERATION BY OPTIMIZER FOR AUTOMATIC QUERY REWRITE, WHICH IS USED FOR DYNAMIC SQL ONLY. TO USE AN MQT IN STATIC SQL, REFER TO THE MQT TABLE NAME DIRECTLY.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User response:</td>
<td>As appropriate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL7027I</th>
<th>*** GOOD NEWS: THIS STATEMENT IS ACCESSING A MATERIALIZED QUERY TABLE (MQT) WHICH SHOULD RESULT IN BETTER, FASTER ACCESS TO YOUR DATA. THE OPTIMIZER CHOSE AN MQT, NOT THE BASE TABLE, USING AUTOMATIC QUERY REWRITE. MQTS ARE READ ONLY TABLES, AND ELAPSED TIME SHOULD BE REDUCED VS. BASE TABLE ACCESS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User response:</td>
<td>As appropriate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL7028I</th>
<th>*** GOOD NEWS: THIS SQL STATEMENT IS RETRIEVING ITS VALUES DIRECTLY FROM BUFFERS IN MEMORY, RESULTING IN QUICKER ACCESS. THIS METHOD OF INSERT WITHIN SELECT IS GOOD FOR RETURNING ROWIDS, IDENTITY COLUMNS, SEQUENCES AND THE RESULTS OF TRIGGERS FROM THE FINAL TABLE.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User response:</td>
<td>As appropriate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANL7029I</th>
<th>*** GOOD NEWS: DYNAMIC SCROLLABLE CURSORS ALLOW BACKWARD INDEX SCAN AND BACKWARD SEQUENTIAL TABLE SCAN, INCLUDING BACKWARD SCANS FOR ORDER BY SORTS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User response:</td>
<td>As appropriate.</td>
</tr>
</tbody>
</table>
**ANL7030I • ANL7038I**

ANL7030I  *** GUIDELINE: AN ASENSITIVE SCROLLABLE CURSOR HAS BEEN DECLARED. IF SQL IS READ ONLY THEN IT BECOMES AN INSENSITIVE CURSOR; IF SQL IS NOT READ ONLY THEN IT BECOMES A SENSITIVE DYNAMIC CURSOR.

User response: As appropriate.

ANL7031I  *** GOOD NEWS: ORDER BY SORT FOR THIS SCROLLABLE CURSOR HAS BEEN AVOIDED USING FORWARD AND/OR BACKWARD SCAN ON A SUITABLE INDEX INSTEAD.

User response: As appropriate.

ANL7032I  *** GUIDELINE: SCROLLABLE CURSORS REQUIRING USE OF WORK FILE CANNOT BE DECLARED SENSITIVE DYNAMIC (USE OF GROUP BY, FOR EXAMPLE). CURSORS WILL NOT REFLECT CHANGES TO SUBQUERY TABLES, AS BASIC PREDICATE IS ONLY EVALUATED FOR OPEN CURSOR. ALSO, BUILT IN FUNCTIONS (MIN, MAX, AVG, COUNT) GET CALCULATED ONLY ONCE AND ARE NOT REEVALUATED; FINALLY, PARALLELISM IS NOT SUPPORTED.

User response: As appropriate.

ANL7033I  *** GUIDELINE: DPSI ARE INDEXES THAT ARE ORDERED BY THE DATA WITHIN EACH PARTITION. THEY ARE GOOD FOR ACCESSING DATA ORGANIZED ALONG PARTITION BOUNDARIES. PARTITION PRUNING MAY OCCUR IF THIS IS DESIGNATED AS THE PARTITIONING INDEX, BUT IF COMPARING INDEX COLUMNS TO HOST VARIABLES OR PARAMETER MARKERS THEN REOPT (VARS) MUST BE USED TO PRUNE. DPSI ARE GOOD WHEN RANDOM DATA ACCESS IS LIMITED, OR WHEN PREDICATES USE COLUMNS FROM BOTH A DPSI AND A TRADITIONAL PARTITIONING INDEX.

User response: As appropriate.

ANL7034I  *** GOOD NEWS: THIS INDEX IS NOT PADDED, MAKING IT A TRUE VARIABLE LENGTH INDEX. INDEX ONLY ACCESS IS NOW POSSIBLE AND DB2 WILL COMPARE CHAR / VARCHAR COLUMNS OF UNEQUAL LENGTH AGAINST THIS INDEX DURING STAGE 1.

User response: As appropriate.

ANL7035I  *** GOOD NEWS: THIS TABLE IS DESIGNATED AS VOLATILE, MEANING THAT DB2 WILL ALWAYS USE AN INDEX TO ACCESS ITS DATA, IF ONE IS AVAILABLE.

User response: As appropriate.

ANL7036I  *** GOOD NEWS: A SPARSE INDEX WAS EMPLOYED BY DB2 TO ACCESS THIS WORK FILE DURING THE STAR JOIN PROCESS. SPARSE INDEXES ARE SUPERIOR TO TABLE SPACE SCANS.

User response: As appropriate.

ANL7037I  *** GUIDELINE: DYNAMIC SQL CAN CACHE THE PREPARED STATEMENT TO REDUCE REEXECUTION OVERHEAD. USE REOPT(ONCE) TO LOAD BEST FIT ACCESS PLAN; USE RUNSTATS WITH REPORTS NO UPDATE NONE TO INVALIDATE (REMOVE) PREPARED SQL STATEMENTS IN CACHE.

User response: As appropriate.

ANL7038I  *** GUIDELINE: INDEX ON EXPRESSION CANNOT BE BUILT ON A SUBQUERY, AGGREGATE FUNCTION, UDF, HOST VARIABLE, PARAMETER MARKER, SPECIAL REGISTER OR CAST EXPRESSIONS. ALSO, DESC ORDER IS NOT PERMITTED.

User response: As appropriate.
Statistics collector messages

**ANL8001I** COLLECTING STATS FOR TABLE  
creator.tablename

**Explanation:** Informational. You have requested collection of all Table, Table space, and Index statistics pertaining to the table identified by its creator (8 characters) and tablename (18 characters). The table must be specified in fixed field A8.A18 format, with a dot or space between the names, as: COLLECT TDT690.L1000; otherwise, the table name cannot be properly identified. Statistics are automatically collected from SYSIBM.SYSTABLES, SYSTABLESPACE and SYSINDEXES.

**User response:** None.

**ANL8002I** CAF OPEN RETURN CODE IS ccc

**Explanation:** Informational. The Call Attach Facility Open routine has returned the value ccc. In normal operations, ccc is 0. A value of 4 or more indicates a problem attaching to the DB2 subsystem. Check the setting of the SUBSYST parameter for accuracy.

**User response:** As appropriate.

**ANL8003I** CAF CLOSE RETURN CODE IS ccc

**Explanation:** Informational. The Call Attach Facility Open routine has returned the value ccc. In normal operations, ccc is 0. A value of 4 or more indicates a problem disconnecting from the DB2 subsystem.

**User response:** As appropriate.

**ANL8004E** ERROR ENCOUNTERED: errcode  
SQLCODE: sqlcode

**Explanation:** Error. The program encountered an error and was unable to continue. The latest error condition (errcode) and DB2 SQLCODE (sqlcode) are returned for diagnosis.

**User response:** As appropriate.

**ANL8005I** PROGRAM COMPLETE, STATISTICS  
COLLECTED FROM:  
SYSIBM.SYSTABLES nnnn  
SYSIBM.SYSTABLESPACE nnnn  
SYSIBM.SYSINDEXES nnnn  
SYSIBM.SYSCOLUMNS nnnn  
SYSIBM.SYSCOLDIST nnnn  
SYSIBM.SYSCOLDISTSTATS nnnn  
SYSIBM.SYSTABSTATS nnnn  
SYSIBM.SYSINDEXSTATS nnnn  
SYSIBM.SYSCOLDISTSTATS nnnn  

**Explanation:** Informational. Final count (nnnn) of the records extracted from each of the potential SYSIBM catalog tables at the end of the program.

**User response:** None.

**ANL8006I** DB2 SUBSYSTEM RELEASE LEVEL IS  
vers.

**Explanation:** Informational. The DB2 version for which output DDL is being written (that is, the migration target).

**User response:** None.

**ANL8007W** THE 200 TABLE COLLECTION LIMIT  
EXCEEDED. ONLY THE FIRST 200  
ENTRIES ARE PROCESSED.

**Explanation:** Warning. You have requested more than the maximum of 200 table entries in a single run. The program stops collecting information after 200 COLLECT parameters have been processed. Those specified continue processing.

**User response:** As appropriate.

**ANL8008I** PARTITION LEVEL STATISTICS ARE  
REQUESTED.

**Explanation:** Informational. Partition level statistics have been requested. ANLCAT collects all entries found in SYSIBM.SYSTABSTATS, SYSINDEXSTATS, SYSCOLSTATS, and SYSCOLDISTSTATS for all partitioned tables collected.

**User response:** None.

**ANL8009W** NO ENTRY FOUND FOR TABLE  
creator.tablename.

**Explanation:** Warning. The specified creator.tablename could not be found in the catalog table SYSIBM.SYSTABLES.

**User response:** Check the setting and spelling of the COLLECT parameter for fixed field (A8.A18), check the SUBSYST parameter for proper subsystem ID, and run again.

**ANL8010I** DB2 SUBSYSTEM NAME USED IS ssid.

**Explanation:** Informational. You have specified the target DB2 subsystem from which ANLCAT collects its catalog statistics.

**User response:** None.

**ANL8011W** NO ENTRY FOUND FOR TABLESPACE  
dbname.tsname.

**Explanation:** Warning. The specified database name (dbname) and table space name (tsname) could not be found in the catalog table SYSIBM.SYSTABLESPACE.
These names were derived from the entry in SYSIBM.SYSTABLES.

User response: Check and correct the catalog, and run again.

ANL8012I  NON-UNIFORM DISTRIBUTION STATS ARE REQUESTED.

Explanation: Informational. You have requested non-uniform column distribution statistics from SYSIBM.SYSCOLDIST to be collected for columns where they exist. This also collects concatenated index key column statistics, if they exist.

User response: None.

ANL8013I  NO INDEX FOUND FOR TABLE creator.tablename.

Explanation: Informational. There were no indexes found in SYSIBM.SYSINDEXES for the table named creator.tablename. Processing continues.

User response: As appropriate.

ANL8014I  NO COLUMN STATS FOR TABLE creator.tablename.

Explanation: Informational. There were no column entries found in SYSIBM.SYSCOLUMNS for the table named creator.tablename. Processing continues.

User response: As appropriate.

ANL8015I  INDIVIDUAL COLUMN LEVEL STATISTICS ARE REQUESTED.

Explanation: Informational. You have requested column level statistics from SYSIBM.SYSCOLUMNS, including COLCARDF, HIGH2KEY, and LOW2KEY.

User response: None.

ANL8016I  NO COLUMN DIST STATS FOR creator.tablename.

Explanation: Informational. There were no non-uniform column distribution statistics found in SYSIBM.SYSCOLDIST for the table named creator.tablename. Processing continues.

User response: As appropriate.

ANL8017I  NO PART COLDIST STATS FOR creator.tablename.

Explanation: Informational. There were no partition-level non-uniform column distribution statistics found in SYSIBM.SYSCOLDISTSTATS for the partitioned table creator.tablename. Processing continues.

User response: As appropriate.

ANL8018I  NO PART COLUMN STATS FOR creator.tablename.

Explanation: Informational. There were no partition-level column statistics (Highkey, High2key, and so on) found in SYSIBM.SYSCOLSTATS for the partitioned table creator.tablename. Processing continues.

User response: As appropriate.

ANL8019I  NO PART TABLE STATS FOR creator.tablename.

Explanation: Informational. There are no statistics in the SYSIBM.SYSTABSTATS table relating to the partitioned table identified by creator.tablename.

User response: As appropriate.

ANL8020I  NO PART INDEX STATS FOR creator.tablename.

Explanation: Informational. There are no statistics in the SYSIBM.SYSINDEXSTATS table relating to the partitioned table identified by creator.tablename.

User response: As appropriate.

ANL8021I  SQLCODE sqlcode UNKNOWN TO DSNTIAR ROUTINE, DB2 HAS NO EXPLAINABLE MESSAGE FOR THIS ERROR.

Explanation: Error. An SQLCODE was returned during processing that is not processed by DB2's DSNTIAR error message routine.

User response: Check the sqlcode in Message and Codes and take appropriate action.

ANL8022I  GLOBAL TARGET CREATOR IS creator.

Explanation: Informational. SQL PA acknowledges that you selected a different creator name to be applied to all output generated by the statistics migration program, rather than propagate the same creator/owner that was defined on the input system.

User response: None.

ANL8023I  GLOBAL TARGET DATABASE NAME IS database.

Explanation: Informational. SQL PA acknowledges that you selected a different database name to be applied to all output generated by the statistics migration program, rather than propagate the same database name as was defined on the input system.

User response: None.
ANL8024I  COLLECTING STATS FOR LONG TABLE authid tabname.

Explanation: Informational. SQL PA acknowledges that you defined a long table name (in Version 8 the names of the creator and table can be up to 128 chars each), specified by separate authorization ID and tabname parameters on the COLLATH and COLLTAB parameters.

ANL8025I  "FROM" DB2 RELEASE LEVEL IS version.

Explanation: Informational message identifies the DB2 level of the source machine.

User response: None.

DB2 SQL Performance Analyzer ISPF panel messages

The following messages are issued under TSO by the SQL PA CLIST and ISPF panel processes:

- **ANL001I**  For statements that contain SQL errors, the explain reports and trace reports do not exist.
  
  Explanation: Explain and trace reports do not exist for statements that contain SQL errors.
  
  User response: No action is required. If you need the explain and trace reports, you must fix any statements that contain SQL errors.

- **ANL002E**  Required values are missing.
  
  Explanation: Some fields are missing required values. The cursor will display in the fields that do not have values.
  
  User response: Specify values for the required fields.

- **ANL003I**  The selected feature is unavailable when the input source is a plan or a package, or when ANLCSPA is used.
  
  Explanation: If a plan or a package was used as the input source or ANLCSPA was used, the selected feature cannot be used.
  
  User response: No action required.

- **ANL004I**  Either the package does not exist or EXPLAIN(YES) is not specified in the BIND command statement.
  
  Explanation: To use COMPARE, the package must exist and EXPLAIN(YES) must not be specified in the BIND command statement.
  
  User response: No action required. To continue, ensure the COMPARE value is set to NO.

- **ANL005E**  The package does not exist in the SYSIBM.SYSPACKAGE catalog table.
  
  Explanation: The specified package does not exist in the catalog table.
  
  User response: No action required. To continue, specify a different package.

- **ANL006E**  The specified value is invalid.
  
  Explanation: An invalid value was specified.
  
  User response: Ensure that a valid value is specified.

- **ANL007E**  The source module was not found.
  
  Explanation: The source module is missing.
  
  User response: Ensure that the ANLCSRCE source module exists.

- **ANL008E**  SQL statements for the source module could not be extracted.
  
  Explanation: Ensure that the ANLCMACR macro was specified on the command line during the ISPF edit session.

- **ANL009E**  The specified data set is unavailable.
  
  Explanation: The specified data set is not available for use.
  
  User response: Ensure that the data set is available, or specify a different data set.

- **ANL010E**  The data set that contains the specified member is not partitioned.
  
  Explanation: When a member was being created, a PDS with a member name was specified.
  
  User response: Either change the data set to a PDS or remove the member name from the data set and use as a sequential data set. If it doesn’t exist, it will be created.

- **ANL011E**  The installed base parameter file was not found.
  
  Explanation: While a new base parameter file was being created, the file that was created by Tools Customizer could not be found.
  
  User response: Restore the missing parameter file.
ANL012E  The specified primary command is invalid
Explanation:  The specified primary command is not valid. Valid primary commands are displayed on the ISPF panel.
User response:  Specify a valid primary command. You can type the command in the command line, or you can put the cursor on the command and press Enter.

ANL013I  Either the plan does not exist or EXPLAIN(YES) is not specified in the BIND command statement.
Explanation:  To use COMPARE, the plan must exist and EXPLAIN(YES) must not be specified in the BIND command statement.
User response:  No action required. To continue, ensure the COMPARE value is set to NO.

ANL014E  The plan does not exist in the SYSIBM.SYSPLAN catalog table.
Explanation:  The specified plan does not exist in the catalog table.
User response:  No action required. To continue, specify a different plan.

ANL015E  Wildcards are not supported. Remove all specified wildcards and try again.
Explanation:  Wildcards, such as % and *, are not supported.
User response:  Remove any wildcard characters.

ANL016E  A DBRM member that matches the specified name or pattern does not exist for this plan.
Explanation:  The specified DBRM member cannot be found.
User response:  Specify a correct name for the DBRM member.

ANL017E  The specified value contains too many characters.
Explanation:  The specified data set name was too long. Data set names can contain up to 44 characters, and each qualifier cannot be more than eight characters.
User response:  Specify a correct name for the DBRM member.

ANL018E  The specified value for the degree of parallelism is invalid. Specify 1 or ANY.
Explanation:  An invalid value for the degree of parallelism was specified. Valid values are 1 and ANY.
User response:  Specify either 1 or ANY.

ANL019E  An alphabetic character value is specified in a field that can contain only a numeric value. Specify a numeric value.
Explanation:  In the QUERYNO field, an alphabetic character instead of a number was specified.
User response:  Specify a numeric value.

ANL020E  The name of the table creator is required. Specify the name of the table creator.
Explanation:  A table name was specified, but the table creator was not specified for the package or plan.
User response:  Specify a table creator.

ANL021E  The specified table does not exist in the plan.
Explanation:  The plan does not contain the specified table.
User response:  Specify an existing table.

ANL022E  The specified table does not exist in the package.
Explanation:  The package does not contain the specified table.
User response:  Specify an existing table.

ANL023E  The specified QUERYNO does exist in the plan.
Explanation:  The plan does not contain the specified QUERYNO.
User response:  Specify an existing QUERYNO.

ANL024E  The specified QUERYNO does exist in the package.
Explanation:  The package does not contain the specified QUERYNO.
User response:  Specify an existing QUERYNO.
ANL025E The specified value of COMPARE is invalid.
Explanation: An incorrect value was specified for COMPARE. Valid values are Yes and No.
User response: To compare the old plan to the new plan, specify Yes. Otherwise, specify No.

ANL026E The specified value of FORCE is invalid.
Explanation: An incorrect value was specified for FORCE. Valid values are Yes and No.
User response: To explain all SQL statements in all packages that conform to the specification, specify Yes. Otherwise, specify No or leave the field blank.

ANL027E The specified pattern was not found in the name of any member.
Explanation: The DBRM member was not found.
User response: Specify a valid DBRM member name.

ANL028E The data set could not be opened.
Explanation: The data set that contains the statements to be explained could not be opened.
User response: Ensure that you have the correct authority to use the data set.

ANL029E The specified data set does not exist or is not cataloged.
Explanation: A data set that does not exist or is not cataloged was specified.
User response: Ensure that the specified data set exists and is cataloged.

ANL030E A member name was not specified for the PDS.
Explanation: A PDS and a member name must be specified together.
User response: Ensure that you specify the member with the PDS.

ANL031E The format of the specified data set is invalid.
Explanation: The data set name was specified in an invalid format.
User response: Ensure that the specified data set name adheres to the following format:
- Contains no more than 44 characters
- Each qualifier contains no more than 8 characters
- Each qualifier is separated by a period (.)

ANL032E The specified qualifier is invalid.
Explanation: The data set name was specified in an invalid format.
User response: Ensure that only one qualifier is specified in the field.

ANL033E The specified QUERYNO was not found in the PLAN_TABLE.
Explanation: The specified QUERYNO does not exist in the PLAN_TABLE.
User response: Ensure that the specified information about the PLAN_TABLE is correct.

ANL034E The QMF query was not found in the OBJECT_DATA table.
Explanation: The OBJECT_DATA table does not contain the specified QMF query.
User response: Change the query name.

ANL035E The specified plan was bound by using the PKLIST option.
Explanation: Plans that were bound by using the PKLIST option must be processed from the DB2 system catalog.
User response: To process plans that were bound from a package, use the option to process packages from the DB2 system catalog.

ANL036E The PLAN_TABLE was not found in the DB2 system catalog for the specified authorization ID.
Explanation: The PLAN_TABLE could not be found.
User response: Ensure that the specified information for the PLAN_TABLE is correct.

ANL037E The table for the specified name and creator was not found in the DB2 system catalog.
Explanation: An incorrect table name was specified.
User response: Ensure that the correct table name is specified.

ANL038E The index for the specified name and creator was not found in the DB2 system catalog.
Explanation: An incorrect index name was specified.
User response: Ensure that the correct table name is specified.
ANL039E  The specified member contains no records.
Explanation:  The member is empty.
User response:  Ensure that the member contains records, or specify a different member name.

ANL040E  The specified member or data set was not found.
Explanation:  The member or data set could not be found.
User response:  Specify a valid member or data set.

ANL041E  The specified index exists in the system catalog.
Explanation:  The index being created exists in the system catalog.
User response:  Specify a different index.

ANL042I  The maximum number of SQL statements that can be explained was reached.
Explanation:  The maximum number of statements that can be explained is 300.
User response:  No action is required. To explain more than 300 statements, specify FORCE=Y.

ANL043E  The specified DB2 run library was not found.
Explanation:  An incorrect DB2 run library was specified.
User response:  Specify a valid DB2 run library.

ANL044E  The specified CEE run library was not found.
Explanation:  An incorrect CEE run library was specified.
User response:  Specify a valid CEE run library.

ANL045E  The specified SQL PA load library was not found.
Explanation:  An incorrect SQL PA load library was specified.
User response:  Specify a valid SQL PA load library.

ANL046E  The specified DB2 load library was not found.
Explanation:  An incorrect DB2 load library was specified.
User response:  Specify a valid DB2 load library.

ANL047E  The specified DB2 exit library was not found.
Explanation:  An incorrect DB2 exit library was specified.
User response:  Specify a valid DB2 exit library.

ANL048E  The specified table space or database was not found in the DB2 system catalog.
Explanation:  An incorrect table space or database was specified.
User response:  Specify a valid table space or database.

ANL049E  The user user_ID is not authorized to change the CURRENT SQLID value to ID.
Explanation:  The specified user does not have the required authority to set the CURRENT SQLID to the specified value.
User response:  Ensure that the specified user has the required authority.

ANL050E  ANLCHECK was not bound.
Explanation:  The SQL ID was set, but ANLCHECK was not bound.
User response:  Submit &ANLSSID.KCHE to bind the ANLCHECK package.

ANL051E  If the value of SQLID is blank, it will be set to +OFF+ after you exit.
User response:  No action is required.

ANL052E  The specified line command is invalid.
Explanation:  The specified line command cannot be used on the panel.
User response:  Specify a valid line command.

ANL053E  In DB2 10 and later releases, all plans must be bound by using the PKLIST option.
Explanation:  The plan was bound by an option that is not supported in DB2 10 and later releases.
User response:  To process plans that were bound from a package, use the option to process packages from the DB2 system catalog.
ANL054E  No explainable statements were found in the selected input.
Explanation: The output contains no explainable statements.
User response: Ensure that the package contains statements that can be explained.

ANL055W  The ANLPARM user parameter file is undefined. Default values will be used for processing, and changes will not be permanently saved.
User response: No action is required.

ANL056W  The ANLCNTL system parameter file is undefined. Default values will be used for processing, and changes will not be permanently saved.
User response: No action is required.

ANL059W  The specified database and table space information cannot be obtained from the SQL_ID.EEE_EEEWORK view. The default database and table space will be used.
User response: If the default database and table space exist, this message is a warning, and no action is required. Otherwise, restore the view and ensure that the required permissions are granted.

ANL060E  A connection to the remote DB2 subsystem was not made. Ensure that SYSIBM.LOCATIONS contains the location_name location.
Explanation: The DB2 subsystem might not be running, or the subsystem might be set up incorrectly.
User response: Ensure that the DB2 subsystem is running and that setup information for the remote subsystem is correct.

ANL061E  Information from the QMF tables could not be retrieved.
Explanation: QMF queries were not found because the specified owner was incorrect.
User response: Ensure that the correct owner is specified, and retrieve the QMF queries again.

ANL062E  The QMF object type must be QUERY.
Explanation: An incorrect QMF object type was specified.
User response: Specify QUERY for the QMF object type.

ANL063E  The QMF query subtype must be SQL.
Explanation: An incorrect QMF query subtype was specified.
User response: Specify SQL for the QMF query subtype.

ANL064E  The QMF query was not saved with SHARE=YES and is restricted to the owner.
Explanation: The user does not have permission to read the QMF query.
User response: Specify SHARE=YES, and ensure that the user has permission to read the QMF query.

ANL065E  The value specified for primary space allocation is not valid.
Explanation: An invalid value for PRQTY was specified. The value must be a nonzero integer that is greater than or equal to -1.
User response: Specify a nonzero integer that is greater than or equal to -1 for the PRQTY parameter.

ANL066E  The value specified for secondary space allocation is not valid.
Explanation: An invalid value for SEQTY was specified. The value must be an integer that is greater than or equal to -1.
User response: Specify an integer that is greater than or equal to -1 for the SEQTY parameter.

ANL067I  VIEW is disabled in the ISPF Configuration Table so BROWSE is used instead.
User response: No action is required. However, consider enabling VIEW so that the correct position can be located for you.

ANL068I  VIEW is disabled in the ISPF Configuration Table so BROWSE is used instead. Query query_number must be found manually.
User response: No action is required. However, consider enabling VIEW so that the specified query number can be located automatically.

ANL069E  During an attempt to view or browse the data_set_name report, a return code of return_code_number was issued.
Explanation: The report could not be viewed or browsed.
User response: Check the return code to determine
### ANL070W • ANL071E

The user **user_ID** is not authorized to save to **parameter_file**. Changes will be effective for only the current session.

**Explanation:** The user ID does not have the required authority to save changes to the parameter file.

**User response:** Ensure that the user has authority to make changes in the parameter file.

### ANL071E

Reports are unavailable because either no previous reports exist or an error occurred when the reports were generated.

**Explanation:** If DB2 SQL Performance Analyzer is being run for the first time, this message is informational. Otherwise, reports were not generated because of an error that occurred while the reports were run.

**User response:** Ensure that the reports exist. If they do not exist, rerun the reports and try to view them again.

### ANL072E

The **SQL_ID.DSN_VIRTUAL_INDEXES** DB2 table could not be created.

**Explanation:** The specified table was not created because the user does not have the authority to create tables.

**User response:** Ensure that the authority to create tables is granted to the user.

### ANL073E

An error occurred while the **data_set_name** data set was being created.

**Explanation:** The user parameter file could not be created because the user does not have the authority to create the file.

**User response:** Ensure that the authority to create the file is granted to the user.

### ANL074E

An error occurred while the parameter values were being copied to the **data_set_name** data set.

**Explanation:** The parameter values could not be copied to the member in the data set because the user does not have the authority to write to the member.

**User response:** Ensure that the authority to write to the member is granted to the user.

### ANL075E

The specified virtual index already exists in the DB2 system catalog.

**Explanation:** The specified virtual index exists.

### ANL076E

The index **index_name** virtual index created by **creator_name** was not found in the DB2 system catalog.

**Explanation:** The specified virtual index could not be found.

**User response:** Verify that the name of the virtual index and the name of the creator are correct.

### ANL077I

The specified index exists in the DB2 system catalog and was already dropped virtually.

**User response:** No action is required.

### ANL078E

The specified index is a virtual index in the DB2 system catalog.

**Explanation:** The specified index is being used as a virtual index so it cannot be used as a real index.

**User response:** Specify a different index.

### ANL079E

The specified column name **column_name** is not in the specified table.

**Explanation:** The specified column name is invalid.

**User response:** Specify a different column name.

### ANL080E

The value must not contain embedded blanks.

**Explanation:** The column name contains blanks.

**User response:** Specify a column name without blanks.

### ANL081E

User **user_ID** needs the privilege for SELECT from **SYSIBM.SYSINDEXES** and **SYSIBM.SYSCOLUMNS**, CREATE INDEX, and CREATE TABLE **owner_name.DSN_VIRTUAL_INDEXES** if **DSN_VIRTUAL_INDEXES** table is not created at SQL PA customization.

**Explanation:** If the **DSN_VIRTUAL_INDEXES** table was not created when DB2 SQL Performance Analyzer was customized, the user needs SELECT privilege to access or create tables.

**User response:** Ensure that the user has the required privilege.
ANL9001  THE DATASET YOU HAVE SELECTED FOR INPUT DOES NOT EXIST.
CHECK YOUR SPELLING AND REENTER THE INPUT DATASET NAME. DATASET NAME NOT FOUND IS: dataset name. THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9001. PLEASE PRESS ENTER TO CONTINUE PROCESSING.

Explanation: The input data set name was not correct, so SQL PA under TSO could not process.
User response: Correct the name and run again.

ANL9001  DATASET SPECIFIED IS NOT A PDS, CANNOT SPECIFY A MEMBER. CHECK YOUR INPUT AGAIN, REENTER THE INPUT DATASET NAME. DATASET NAME NOT FOUND IS: dataset(member).

Explanation: The input data set that you specified is not a PDS, so the member name is invalid.
User response: Verify the names of your input data set and member, and try again.

ANL9001  DATASET MEMBER NAME SELECTED FOR INPUT DOES NOT EXIST. CHECK YOUR INPUT AGAIN, REENTER THE DATASET AND MEMBER. DATASET NAME NOT FOUND IS: dataset(member).

Explanation: The member name that you specified for input does not exist in the PDS library.
User response: Verify the names of your input data set and member, and try again.

ANL9002  THE TARGET HOST SYSTEM YOU HAVE SELECTED DOES NOT EXIST. CHECK YOUR SPELLING AND REENTER THE TARGET HOST DB2 NAME. DATASET NAME NOT FOUND IS: sysx.ANLvxxvv.SANLDATA(target) THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9002 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

Explanation: The host system is not a member of the host configuration parameters library, ANLvxxvv.SANLDATA, so the configuration is not selectable, or the member was not added to the ANLPARM2 panel (&QCNTL variable).
User response: If SQL PA allowed you to specify this name as a parameter, then notify system support, because the ANLPARM panel and target hosts data set are not in synchronization. Correct the name and run again.

ANL9003  ANSQLPA ENCOUNTERED PROCESSING ERRORS DURING EXECUTION. THE LAST KNOWN CONDITION CODE RETURN WAS return-code SQL PA MUST TERMINATE THIS RUN. NOTIFY SQL PA SYSTEM SUPPORT. THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9003 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

Explanation: ANSQLPA could not start properly. Possible causes are incorrect SUBSYS'T name for DB2 or DB2 could be down, in which case the call attachment facility does not respond.
User response: Check condition code and try again.

ANL9004  ANSQLPA ENCOUNTERED PROCESSING ERRORS DURING EXECUTION. THE LAST KNOWN CONDITION CODE RETURN WAS return-code SQL PA MUST TERMINATE THIS RUN. NOTIFY SQL PA SYSTEM SUPPORT. THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9004 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

Explanation: ANSQLPA could not start properly. Possible causes are incorrect SUBSYS'T name for DB2 or DB2 could be down, in which case the call attachment facility does not respond.
User response: Check condition code and try again.

ANL9005  THE REPORT YOU HAVE SELECTED IS NOT CURRENTLY AVAILABLE. THE COST SUMMARY LOG SHOULD ALWAYS BE SO. CALL SUPPORT. SQL PA CANNOT LOCATE THE FILE: userid.ANLCOST.LOG THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9005 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

Explanation: The ANLCOST.LOG data set, created under the running TSO user ID, as disappeared.
**ANL9005 • ANL9011**

**User response:** Back out of SQL PA, allocate a new file, and run again.

---

**ANL9005**

AN ALLOCATION ERROR OCCURRED ON SQL PA COST REPORT FILE. THE LAST KNOWN CONDITION CODE RETURNED WAS return code. SQL PA CANNOT ALLOCATE THE FILE: userid.ANLCOSt.LOG THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9005 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

**Explanation:** This version of ANL9005 detects the rare case when the ANLCOST.LOG data set, created under the running TSO user ID, cannot be allocated because it already exists but is not cataloged, on one or more volumes. The file is required for SQL PA operation under TSO.

**User response:** In native TSO mode (not ISPF), enter:

ALLOC DS(ANLCOST.LOG) NEW SP(1,1) TR

This results in a catalog error, along with identification of the disk volume where the uncataloged data set resides. Then, use ISPF 3.2 panel to catalog the data set on that volume, and run SQL PA again.

---

**ANL9006**

THE REPORT YOU HAVE SELECTED IS NOT CURRENTLY AVAILABLE. REPORT OPTIONS REP, TWO, EXP, OR TRE, ALL, DET WILL PRODUCE THIS RPT. SQL PA CANNOT LOCATE THE FILE: data set name THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9006 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

**Explanation:** The SQL PA Enhanced Explain report data set, whose name you specified on the SQL PA Select Output Reports panel, cannot be located.

**User response:** Back out of SQL PA, allocate a new file, and run again.

---

**ANL9007**

THE REPORT YOU HAVE SELECTED IS NOT CURRENTLY AVAILABLE. ONLY REPORT OPTIONS DET, TRE, OR ALL WILL PRODUCE THIS RPT. SQL PA CANNOT LOCATE THE FILE: data set name THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9007 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

**Explanation:** The SQL PA Detail Trace report data set that you specified on the SQL PA Select Output Reports panel, cannot be located.

**User response:** Back out of SQL PA, allocate a new file, and run again.

---

**ANL9008**

THE REPORT YOU HAVE SELECTED IS NOT CURRENTLY AVAILABLE. REPORT OPTIONS REP, TWO, EXP, OR TRE, ALL, DET WILL PRODUCE THIS RPT. SQL PA CANNOT LOCATE THE FILE: data set name THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9008 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

**Explanation:** The SQL PA Limits report data set that you specified on the SQL PA Select Output Reports panel, cannot be located.

**User response:** Exit from SQL PA and issue the request again, a new report is allocated.

---

**ANL9009**

EDIT INPUT DATA SET IS NOT CURRENTLY AVAILABLE TO SQL PA. DATA SET OR MEMBER MAY NOT EXIST, OR COULD BE ALREADY IN USE OR EDIT BY ANOTHER USER:

dataset-name THE SQL PA ERROR CODE SET BY THIS MESSAGE IS: 9009 PLEASE PRESS ENTER TO CONTINUE PROCESSING.

**Explanation:** The data set chosen for input is being edited by another user or does not exist.

**User response:** Retry later, or choose another input source.

---

**ANL9010**

THE INPUT SQL DATASET FOR COMPARE FUNCTION DOES NOT EXIST. EASY EXPLAIN CREATES THIS SQL FILE USING TOPA=ALL OPTION SQL PA CANNOT ALLOCATE OR FIND DATASET: edsn

**Explanation:** The data set name from the compare old and new plans panel is not a valid name, or cannot be located. This file contains the SQL statements that match the entries in EEPATH table for comparisons.

**User response:** Check the spelling to verify that you specified the data set name correctly. Access the Easy Explain function, and create the ANLLEEE.SQL or equivalent input data set.

---

**ANL9011**

THE REPORT YOU HAVE SELECTED IS NOT CURRENTLY AVAILABLE. THE DB2 SQL AND OBJECTS FILE SHOULD ALWAYS BE ALLOCATED. NOTIFY SYSTEMS SUPPORT FOR MISSING: userid.ANLOUT.SQL

**Explanation:** The userid.ANLOUT.SQL data set must be created by the CLISTS.

**User response:** Contact IBM Software Support to report the missing data set.
ANL9012  THE DATA SET OR MEMBER TO BE
RESTORED DOES NOT EXIST CHECK
YOUR SPELLING FOR THIS MEMBER
NAME AND THEN RETRY. SQL PA
CANNOT LOCATE:
userid.ANL410.SANLSTAT (member)

Explanation: The data set member requested to
supply parameters for the RESTORE function cannot be
found. Under each TSO user ID, a repository of
statistics is built and maintained in
userid.ANL410.SANLSTAT for What If? and Statistics
Migration online.

User response: Check the spelling to verify that you
specified the member name correctly. Issue the
following TSO command to verify the members:
LISTD ANL410.SANLSTAT ME

ANL9013  SQL PA ENCOUNTERED PROBLEMS
RESTORING CATALOG STATISTICS
FROM userid.ANL410.SANLSTAT
(member), RC: retcode CHECK RETURN
CODE, AND CONSIDER MANUAL
STATS RESTORATION

Explanation: The What If? Analysis (ANLWIF)
program was unable to use the member that was
requested for the RESTORE function or it had problems
restoring the statistics.

User response: Run the TSO online interface from
ISPF 6, and specify ANLI QDEBUG(YES). All error
messages from ANLWIF are routed to SYSPRINT
(screen). Check the SQL CODE for type of error
reported and correct the problem. Possible SQL CODE
is –545, a cluster ratio not between 0 and 1.

ANL9014  SQL PA ENCOUNTERED PROBLEMS
RETRIEVING CATALOG STATISTICS
FOR DISPLAY. CHECK SPELLING OF
YOUR CREATOR AND INDEX NAME
AND ALSO THE SUBSYSTEM ID,
THEN RETRY. RC: retcode

Explanation: The What If? Analysis (ANLWIF)
program encountered problems while trying to access
an index's statistics.

User response: Run the TSO online interface from
ISPF 6, and specify ANLI QDEBUG(YES). All error
messages from ANLWIF are routed to SYSPRINT
(screen). Check the SQL CODE for type of error
reported and correct the problem.

ANL9015  SQL PA ENCOUNTERED PROBLEMS
UPDATING CATALOG STATISTICS
FOR THIS TABLE. CONTACT SYSTEM
SUPPORT. RC: retcode SQL PA WAS
UNABLE TO UPDATE THE CATALOG
FOR THIS TABLE.

Explanation: The What If? Analysis (ANLWIF)
program encountered problems while trying to modify
catalog statistics for a table.

User response: Run the TSO online interface from
ISPF 6, and specify ANLI QDEBUG(YES). All error
messages from ANLWIF are routed to SYSPRINT
(screen). Check the SQL CODE for the type of error
reported and correct the problem.

ANL9016  SQL PA ENCOUNTERED PROBLEMS
CREATING A NEW INDEX IN THE
DB2 CATALOG. CONTACT SYSTEM
SUPPORT. RC: retcode SQL PA IS
UNABLE TO CREATE AN INDEX ON
SUBSYSTEM ssid

Explanation: The What If? Analysis (ANLWIF)
program encountered problems while trying to create a
new entry in the catalog for an index.

User response: Run the TSO online interface from
ISPF 6, and specify ANLI QDEBUG(YES). All error
messages from ANLWIF are routed to SYSPRINT
(screen). Check the SQL CODE for the type of error
reported and correct the problem. Possible SQL CODE
is –545, a cluster ratio not between 0 and 1.
Tools Customizer messages

These topics describe the error messages and return codes that are issued by Tools Customizer.
Tools Customizer messages

Use the information in these messages to help you diagnose and solve Tools Customizer problems.

---

**CCQB000I** The product parameter data was saved in the data store.

**Explanation:** Changes that were made to the product parameters were saved in the data store.

**System action:** None.

**User response:** No action is required.

---

**CCQB001I** The DB2 parameter data was saved in the data store.

**Explanation:** Changes that were made to the DB2 parameters were saved in the data store.

**System action:** None.

**User response:** No action is required.

---

**CCQB002I** The LPAR parameter data was saved in the data store.

**Explanation:** Changes that were made to the LPAR parameters were saved in the data store.

**System action:** None.

**User response:** No action is required.

---

**CCQB003E** At least one step must be selected in a selected task. The selected task is task_description.

**Explanation:** When a task is selected, at least one step must be selected. A selected step is missing from the specified task.

**System action:** Processing stops.

**User response:** Select a step in the specified task or deselect the task.

---

**CCQB004I** The required information to run the Discover EXEC was saved in the data store.

**Explanation:** The data store contains all the information that is required to run the Discover EXEC.

**System action:** None.

**User response:** No action is required.

---

**CCQB005E** The conflicting values for the parameter_name parameter must be resolved before the information can be saved.

**Explanation:** Two values for one parameter conflict with each other, and they must be resolved to save the information.

**System action:** Processing stops.

**User response:** Resolve the conflicting values for the parameter.

---

**CCQB006E** One row must be selected.

**Explanation:** One row in the table must be selected.

**System action:** Processing stops.

**User response:** Select one row.

---

**CCQB007E** Only one row can be selected.

**Explanation:** Multiple rows in the table are selected, but only one row is allowed to be selected.

**System action:** Processing stops.

**User response:** Select only one row.

---

**CCQC000I** The jobs have been customized on the selected DB2 entries.

**Explanation:** The jobs were customized on the DB2 entries that were selected.

**System action:** None.

**User response:** Press Enter to clear the message.

---

**CCQC001W** The jobs were not generated on one or more of the selected DB2 entries. Press PF3 to check the DB2 entries that were not customized.

**Explanation:** The product was not customized on one or more of the DB2 entries that were selected.

**System action:** None.

**User response:** Press PF3 to see the DB2 entries on which the product was not customized. The status of these DB2 entries is Errors in Customization.

---

**CCQC002I** The edit session was started automatically because values for required parameters are missing or must be verified.

**Explanation:** If product, LPAR parameters, or DB2 parameters are not defined or if parameter definitions must be verified, an editing session for the undefined or unverified parameters starts automatically.

**System action:** None.

**User response:** Define values for all required product,
LPAR parameters, or DB2 parameters.

CCQC003W The template_name template in the library_name metadata library does not contain any parameters.

Explanation: The specified template does not have parameters.

System action: None.

User response: No action is required.

CCQC004S The value of the "type" attribute for the template_name template in the library_name metadata library does not match the value that was previously specified. The value is value_name, and the previously specified value is value_name.

Explanation: The value of the "type" attribute must match the value that was previously specified.

System action: Processing stops.


CCQC005S The template_name template exceeds the number of allowed templates for a customization sequence. The template is in the library_name metadata library.

Explanation: The customization sequence can process only number templates. The specified template cannot be processed because the customization sequence already contains the maximum number of templates.

System action: Processing stops.


CCQC006E The jobs could not be generated for the group_attach_name DB2 group attach name.

Explanation: The customization jobs could not be generated for the specified DB2 group attach name.

System action: Processing stops.


CCQC007E The jobs could not be generated for the subsystem_ID DB2 subsystem.

Explanation: The customization jobs could not be generated for the specified DB2 subsystem.

System action: Processing stops.


CCQC008E The jobs could not be generated for the member_name DB2 member.

Explanation: The customization jobs could not be generated for the specified DB2 member.

System action: Processing stops.


CCQC009S The jobs were not generated for the DB2 entries.

Explanation: One or more errors occurred while customization jobs were being generated for the selected DB2 entries.

System action: Processing stops.


CCQC010S The template_name template could not be accessed in the library_name metadata library.

Explanation: The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action: Processing stops.

User response: Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

CCQC011S The template_name template could not be written to the library_name customization library.

Explanation: The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action: Processing stops.
User response: Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

CCQC012W  The job card was generated with default values because the JOB keyword was missing.

Explanation: Default values were used to generate the job card because the JOB keyword was not specified in the first line of the job card.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add the JOB keyword in the first line of the job card.

CCQC013W  The job card was generated with the default value for the programmer name because the specified programmer name exceeded 20 characters.

Explanation: Default values were used to generate the job card because the specified programmer name contained too many characters.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add a valid programmer name in the job card. A valid programmer name is 1 - 20 characters.

CCQC014W  The job card was generated with default values because the JOB keyword was not followed by a space.

Explanation: Default values were used to generate the job card because a space did not follow the JOB keyword.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add a space after the JOB keyword in the job card.

CCQC015S  The template_name template in the library_name metadata library contains the following file-tailoring control statement: statement_name. This control statement is not valid in a template_type template.

Explanation: The template_type template cannot contain the specified type of file-tailoring control statement.


CCQC016S  The )DOT file-tailoring control statement exceeded the number of allowed occurrences for the template_name template in the library_name metadata library.

Explanation: The )DOT file-tailoring control statement can occur only a limited number of times in the specified template.

System action: Processing stops.


CCQC017S  The nested )DOT file-tailoring control statements exceeded the number of allowed occurrences in the template_name template in the library_name metadata library.

Explanation: Nested )DOT file-tailoring control statements can occur only number times.

System action: Processing stops.


CCQC018S  The template_name template in the library_name metadata library is not valid because it does not contain any data.

Explanation: The specified template is missing required data.

System action: Processing stops.


CCQC019S  The template_name template in the library_name metadata library is not valid because an )ENDDOT file-tailoring control statement is missing.

Explanation: A )ENDDOT file-tailoring control statement is required in the specified template.

System action: Processing stops.

CCQC021S  The template_name template in the library_name metadata library is not valid because the template must start with the parameter_name job card parameter.

Explanation:  The specified template must start with the specified job card parameter.

System action:  Processing stops.


CCQC022S  The parameters used in a )DOT file-tailoring control statement exceeded the number of allowed parameters in the template_name template. The template is in the library_name metadata library. The error occurs in )DOT section section_number.

Explanation:  A )DOT file-tailoring control statement can contain only a limited number of parameters.

System action:  Processing stops.


CCQC023S  The )DOT file-tailoring control statement must include the table-name table name in the template_name template. The template is in the library_name metadata library. The error occurs in )DOT section section_number.

Explanation:  The )DOT file-tailoring control statement is missing a required table name.

System action:  Processing stops.


CCQC024S  ISPF file tailoring failed for the template_name template in the library_name metadata library.

Explanation:  An error occurred during ISPF file tailoring for the specified template.

System action:  Processing stops.

User response:  Review the Tools Customizer-generated trace data set and the ISPF file tailoring trace data set. To create an ISPF file tailoring trace data set, complete the following steps:
1. Run Tools Customizer until the error is about to occur.
2. Specify the ISPFTTRC command, and press Enter.
3. Issue the Tools Customizer command that issues the error.
4. Specify the ISPFTTRC command, and press Enter. The ISPF file tailoring trace data set is created. It adheres the following naming convention: TSO_ID.ISPFT.TRACE, where TSO_ID is the TSO user ID that is being used.

If the problem persists, gather the following information and contact IBM Software Support.
• A screen capture of the Tools Customizer error. Ensure that the complete error message is displayed by pressing PF1.
• The Tools Customizer trace data set. It adheres to the following naming convention: TSO_ID.CCQ.TRACE, where TSO_ID is the TSO user ID that is running Tools Customizer.
• The ISPF file tailoring trace data set.

CCQC025I  Customized jobs do not exist because they have not been generated.

Explanation:  The list of customized jobs cannot be displayed because the product has not been customized for any DB2 entries.

System action:  None.

User response:  Complete the steps to customize a product. Customized jobs are generated when all required product, LPAR parameters, and DB2 parameters are defined and at least one DB2 entry on which to customize the product has been selected.

CCQC026S  The value of the "customized" attribute for the parameter_name parameter in the library_name metadata library template does not match the value that was previously specified. The value is value_name, and the previously specified value is value_name.

Explanation:  The value for the "customized" attribute for a parameter must match the value that was previously specified.

System action:  Processing stops.


CCQC027S  The job_name customization job was not found in the library_name customization library.

Explanation:  The selected customization job does not exist in the customization library.

System action:  Processing stops.

CCQC028S  The library_name customization library was not found.

Explanation:  The customization library does not exist.
System action:  Processing stops.

CCQC029I  The customization jobs were generated for Product_name.

Explanation:  The customization jobs were generated for the specific product.
System action:  None.
User response:  No action is required.

CCQC030S  The customization jobs cannot be generated because at least one DB2 entry must be associated with this product.

Explanation:  The product that you are customizing requires at least one DB2 entry to be associated with it before customization jobs can be generated.
System action:  None.
User response:  Associate a DB2 entry with the product that you are customizing, and regenerate the jobs.

CCQC031I  The jobs were generated for the associated DB2 entries.

Explanation:  The customization jobs were generated for the DB2 entries that are associated with the product.
System action:  None.
User response:  No action is required.

CCQC032S  The customization jobs were not generated for Product_name.

Explanation:  A severe error occurred while the jobs were being generated for the specified product.
System action:  None.

CCQC033S  The customization_library_name has no customized jobs.

Explanation:  The specified customization library cannot be browsed or edited because it is empty.
System action:  None.
User response:  Generate customization jobs for the specified library, and browse or edit the library again.

CCQC034S  The specified operation is not allowed.

Explanation:  Issuing commands against customization jobs from the customization library from an ISPF browse or edit session that was started on the Finish Product Customization panel is restricted.
System action:  None.
User response:  To make changes to customization jobs, follow the steps for recustomization.

CCQC035E  Before you generate customization jobs, edit the product parameters to select one or more tasks or steps, and then issue the G line command or the GENERATEALL command again.

Explanation:  One or more tasks or steps must be selected before customization jobs can be generated.
System action:  None.
User response:  Edit the product parameters to select one or more tasks or steps. Then, issue the G line command or the GENERATEALL command again.

CCQC036E  Before you exit the Product Parameters panel, you must select one or more tasks or steps to generate customization jobs or issue the CANCEL command.

Explanation:  One or more tasks or steps must be selected to generate customization jobs or the CANCEL command must be issued before you can exit the Product Parameters panel.
System action:  None.
User response:  Select one or more tasks or steps, or issue the CANCEL command.

CCQD000W  The member_name environment index member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:  While determining if the specified environment index member is valid, the PL/I XML parser issued an exception warning code.
System action:  Processing continues.
User response:  See the Enterprise PL/I for z/OS
CCQD001S  
The member_name environment index member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the specified environment index member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the error.

CCQD002S  
The XML structure of the member_name environment index member is not valid. The element_name element is unknown.

Explanation: The specified environment index member contains an unknown element.

System action: Processing stops.


CCQD003S  
The XML structure of the member_name environment index member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.


CCQD004S  
The XML structure of the member_name environment index member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.


CCQD005S  
The XML structure of the member_name environment index member is not valid. The content length for the element_name element exceeds maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQD006S  
The XML structure of the member_name environment index member is not valid. The element_name element cannot occur more than maximum_number times.

Explanation: The specified element occurs too many times in the environment index member.

System action: Processing stops.


CCQD007S  
The XML structure of the member_name environment index member is not valid. The element_name element must occur at least minimum_number times.

Explanation: The specified element does not occur enough times in the environment index member.

System action: Processing stops.


CCQD008S  
The XML structure of the member_name environment index member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

Explanation: The specified attribute occurs too many times in the environment index member.

System action: Processing stops.


CCQD009S  
The XML structure of the member_name environment index member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.
CCQD010S  The XML structure of the member_name environment index member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: Content was found in an attribute that cannot contain content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.


CCQD011S  The XML structure of the member_name environment index member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: An attribute does not contain required content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.


CCQD012S  The XML structure of the member_name environment index member is not valid. The content length for the element_name element exceeds maximum_number characters.

Explanation: An element contains too many characters. The name of the element and the maximum number of allowed characters are indicated in the message text.

System action: Processing stops.


CCQD020S  CCQD050S

CCQD050S  The following LPAR serial number is duplicated in the environment index member: serial_number.

Explanation: The environment index member contains duplicate LPAR serial numbers. The duplicate serial number is indicated in the message text.

System action: Processing stops.


CCQD051S  The following DB2 serial number is duplicated in the environment index member: serial_number.

Explanation: The environment index member contains duplicate DB2 serial numbers. The duplicate serial number is indicated in the message text.

System action: Processing stops.


CCQD052S  The following DB2 group attach name is duplicated in the environment index member: group_attach_name.

Explanation: The environment index member contains duplicate group attach names.

System action: Processing stops.


CCQD053S  The reference to the following DB2 subsystem for a DB2 group attach name is duplicated in the environment index member: subsystem_ID.

Explanation: The environment index member contains

CCQD054S  The reference to the following DB2 subsystem for the LPAR name LPAR is duplicated in the environment index member: subsystem_ID.

Explanation: The environment index member contains duplicate references to a DB2 subsystem for an LPAR. The duplicate subsystem ID is indicated in the message text.

System action: Processing stops.

CCQD055S  The following DB2 group attach name was not found in the environment index member: group_attach_name.

Explanation: A group attach name that is referenced by a DB2 member does not exist in the environment index member.

System action: Processing stops.

CCQD056S  The following LPAR was not found in the environment index member: LPAR_name.

Explanation: The LPAR does not exist in the environment index member.

System action: Processing stops.

CCQD057S  The following LPAR is duplicated in the environment index member: LPAR_name.

Explanation: The environment index member contains duplicate LPARs. The name of the duplicate LPAR name is indicated in the message text.

System action: Processing stops.

CCQD100W The member_name product index member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the product index member is valid, the PL/I XML parser issued the specified exception warning code.

System action: Processing continues.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception warning code.

CCQD101S The member_name product index member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the product index member is valid, the PL/I XML parser issued the specified exception error code. Ensure that the Tools Customizer data store data set DCB is the same as the sample SCCQSAMP(CCQCDATS) data set DCB.

CCQD102S The XML structure of the member_name product index member is not valid. The element_name element is unknown.

Explanation: The specified product index member contains an unknown element.

System action: Processing stops.

CCQD103S The XML structure of the member_name product index member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: Content was found for an element that cannot contain content.

System action: Processing stops.
CCQD104S The XML structure of the member_name product index member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.


CCQD105S The XML structure of the member_name product index member is not valid. The content length for the element_name element exceeds maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQD106S The XML structure of the member_name product index member is not valid. The element_name element cannot occur more than maximum_number times.

Explanation: The specified element occurs too many times in the product index member.

System action: Processing stops.


CCQD107S The XML structure of the member_name product index member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: An attribute does not occur enough times in the product index member.

System action: Processing stops.


CCQD108S The XML structure of the member_name product index member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

Explanation: An attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.


CCQD109S The XML structure of the member_name product index member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute does not occur enough times in the product index member.

System action: Processing stops.


CCQD110S The XML structure of the member_name product index member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: An attribute cannot contain content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.


CCQD111S The XML structure of the member_name product index member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: An attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.


CCQD112S The XML structure of the member_name product index member is not valid. The content length for the element_name element exceeds maximum_number characters.
CCQD113S  The XML structure of the member_name product index member is not valid. The attribute_name attribute in the element_name element is unknown.
Explanation: The specified attribute in the product index member is unknown.
System action: Processing stops.

CCQD118S  The content of the member_name product index member is not valid. The configuration_ID configuration ID for the configuration-name configuration name is not unique.
Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the following exception warning code: code_number.
System action: Processing continues.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception warning code.

CCQD120S  The content of the member_name product index member is not valid. The pack ID pack_ID that is referenced by product prefix product_prefix in the metadata library library_name could not be found.
Explanation: The specified pack ID could not be found in the metadata library.
System action: Processing stops.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception error code.

CCQD121I  The specified component metadata library was previously specified as part of the pack_name.
Explanation: The specified metadata library for the component was previously specified as part of a pack.
System action: None.
User response: No action is required.

CCQD123E  The customization library name library_name is being used by another product or component. Specify another customization library qualifier on the Tools Customizer Settings panel.
Explanation: A different product or component is using the specified customization library.
System action: None.
User response: Specify another customization library qualifier on the Tools Customizer Settings panel.

CCQD300W  The member_name product environment member is not valid. The PL/I XML parser issued the following exception warning code: code_number.
Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the specified exception warning code.
System action: Processing continues.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception warning code.

CCQD301S  The member_name product environment member is not valid. The PL/I XML parser issued the following exception error code: code_number.
Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the specified exception error code.
System action: Processing stops.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception error code.

CCQD302S  The XML structure of the member_name product environment member is not valid. The element_name element is unknown.
Explanation: The specified product environment member contains an unknown element.
System action: Processing stops.
CCQD303S  The XML structure of the member_name product environment member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: Content was found for an element that cannot contain content.

System action: Processing stops.


CCQD304S  The XML structure of the member_name product environment member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.


CCQD305S  The XML structure of the member_name product environment member is not valid. The content length for the element_name element exceeds maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQD306S  The XML structure of the member_name product environment member is not valid. The element_name element cannot occur more than maximum_number times.

Explanation: The specified element occurs too many times in the product environment member.

System action: Processing stops.


CCQD307S  The XML structure of the member_name product environment member is not valid. The element_name element must occur at least minimum_number times.

Explanation: The specified element does not occur enough times in the product environment member.

System action: Processing stops.


CCQD308S  The XML structure of the member_name product environment member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

Explanation: The specified attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.


CCQD309S  The XML structure of the member_name product environment member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute does not occur enough times in the product environment member.

System action: Processing stops.


CCQD310S  The XML structure of the member_name product environment member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

The XML structure of the member_name product environment member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.


The XML structure of the member_name product environment member is not valid. The content length for the element_name element exceeds maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


The XML structure of the member_name product environment member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation: The specified attribute in the product environment member is unknown.

System action: Processing stops.


The subsystem_ID DB2 subsystem is associated with this product.

Explanation: The specified DB2 subsystem was added and saved in the Tools Customizer data store for the product to be customized.

System action: Processing continues.

User response: No action is required.

The group_attach_name DB2 group attach name is associated with this product.

Explanation: The specified DB2 group attach name was added and saved in the Tools Customizer data store for the product to be customized.

System action: Processing continues.

User response: No action is required.

The group_attach_name DB2 group attach name is already associated with this product.

Explanation: The specified DB2 group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action: None.

User response: Ensure that the DB2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

The member_name DB2 member for the group_attach_name DB2 group attach name is already associated with this product.

Explanation: The specified DB2 member for the group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action: None.

User response: Ensure that the DB2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.
CCQD356S  The library_name metadata library is already associated with the maximum number of allowed DB2 entries for this product.

Explanation: The specified metadata library cannot be associated with more DB2 entries because it is already associated with the number of DB2 entries that are allowed.

System action: Processing stops.

User response: Delete an associated DB2 entry, and associate the specified library with another DB2 entry again.

CCQD357I  The subsystem_ID DB2 subsystem is unassociated with this product.

Explanation: The specified DB2 SSID was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD358I  The member_name DB2 member for the group_attach_name DB2 group attach name is unassociated with this product.

Explanation: The specified DB2 member for the DB2 group attach name was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD359I  The group_attach_name DB2 group attach name is unassociated with this product.

Explanation: The specified DB2 group attach name was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD360S  The library_name metadata library is not associated with the specified DB2 subsystem subsystem_ID.

Explanation: The specified DB2 subsystem and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 subsystem and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD361S  The library_name metadata library is not associated with the specified DB2 data sharing group member member_name for the group_attach_name DB2 group attach name.

Explanation: The specified DB2 data sharing group member for the group attach name and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 data sharing group member for the group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD362S  The library_name metadata library is not associated with the specified group_attach_name DB2 group attach name.

Explanation: The specified DB2 group attach name and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD400W  The customization parser issued the code_number warning code while it parsed the product customization member member_name. See the PL/I programming guide for more information about this XML parser continuable exception code.

Explanation: While determining if the specified member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing stops.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

CCQD401S  The customization parser issued the code_number error code while it parsed the product customization member member_name. See the PL/I programming guide for more information about this XML parser terminating exception code.

Explanation: While determining if the specified member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the Enterprise PL/I for z/OS
CCQD500W  The data_set_name data store data set was not found.
Explanation:  Tools Customizer could not find the specified data store data set.
System action:  None.
User response:  No action is required.

CCQD501W  The data_set_name data store data set was not found, so it was created.
Explanation:  Tools Customizer created the specified data set because it could not be found.
System action:  None.
User response:  No action is required.

CCQD502E  The data_set_name data store data set is not writable.
Explanation:  Tools Customizer cannot write to the specified data set.
System action:  None.
User response:  Ensure that the data set is writable.

CCQD503E  The data_set_name data store data set could not be opened with the disposition_type disposition.
Explanation:  Tools Customizer could not open the data set with the specified disposition.
System action:  Processing stops.
User response:  Ensure that you have WRITE authority access to this data set.

CCQD504E  The data_set_name data store data set could not be opened with the option_name option.
Explanation:  Tools Customizer could not open the data set with the specified option.
System action:  Processing stops.
User response:  Ensure that you have WRITE authority access to this data set.

CCQD505E  The data_set_name data store data set could not be created.
Explanation:  Tools Customizer could not create the specified data set.
System action:  Processing stops.
User response:  Ensure that you have the authority to create data sets and that the DASD is not full.

CCQD510I  The DB2 SSID and DB2 group attach name were created.
Explanation:  The DB2 SSID and DB2 group attach name were created and saved in the data store.
System action:  None.
User response:  No action is required.

CCQD511E  The DB2 entry already exists in the list of DB2 entries to be associated.
Explanation:  The DB2 entry cannot be added because it already exists in the list of DB2 entries to be associated.
System action:  None.
User response:  Specify a different DB2 entry.

CCQD512S  An error occurred while a DB2 entry was being created.
Explanation:  A severe error occurred while a DB2 entry was being created.
System action:  Processing stops.

CCQD513E  The specified DB2 entry already exists and is associated with the current product on the Customizer Workplace panel.
Explanation:  The DB2 entry cannot be added because it already exists, and it is already associated with the product to be customized.
System action:  None.
User response:  Press F3 to go to the Customizer Workplace panel to see the DB2 entry, or specify a different DB2 entry.

CCQD514E  A value is required for a DB2 subsystem, a DB2 group attach name, or both before they can be created.
Explanation:  Required information is missing. A DB2 subsystem, a DB2 group attach name, or both must be specified.
System action:  None.
User response:  Specify a DB2 subsystem, a DB2 group attach name, or both.
**CCQD515E** The specified DB2 entry already exists in the list of DB2 entries and is already associated with the current product.

**Explanation:** The DB2 entry has already been created and associated with the product that you want to customize.

**System action:** None.

**User response:** Specify a different DB2 entry.

---

**CCQD516E** The specified DB2 entry already exists in the list of DB2 entries on the Associate DB2 Entry with Product panel but is not associated with the current product.

**Explanation:** The DB2 entry exists, but it must be associated with the product to be customized.

**System action:** None.

**User response:** On the Customizer Workplace panel, issue the ASSOCIATE command to associate the DB2 entry with the product.

---

**CCQD517S** An error occurred while a DB2 entry was being copied.

**Explanation:** A severe error occurred while a DB2 entry was being copied.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

**CCQD518E** A value is required for a DB2 subsystem, a DB2 group attach name, or both before they can be copied.

**Explanation:** Required information is missing. A DB2 subsystem, a DB2 group attach name, or both must be specified.

**System action:** None.

**User response:** Specify a DB2 subsystem, a DB2 group attach name, or both.

---

**CCQD519I** The DB2 entry was copied.

**Explanation:** The DB2 entry was copied and saved in the Tools Customizer data store.

**System action:** None.

**User response:** No action is required.

---

**CCQD520S** The DB2 entry was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

**Explanation:** The DB2 entry was not completely copied because a product can be associated with only 1200 DB2 entries.

**System action:** Processing stops.

**User response:** Remove a DB2 entry from the list, and copy the specified DB2 entry again.

---

**CCQD521E** Line command is not a valid line command.

**Explanation:** The specified line command is not valid. Valid line commands are on the panel.

**System action:** Processing stops.

**User response:** Specify a valid line command.

---

**CCQD522E** The subsystem_ID DB2 subsystem ID occurs more than once in the list. Each row must be unique.

**Explanation:** The specified DB2 subsystem ID can be used only once.

**System action:** Processing stops.

**User response:** Specify a different DB2 subsystem ID.

---

**CCQD523E** The group_attach_name DB2 group attach name occurs more than once in the list. Each row must be unique.

**Explanation:** The specified DB2 group attach name can be used only once.

**System action:** Processing stops.

**User response:** Specify a different DB2 group attach name.

---

**CCQD524E** The member_name DB2 member for the DB2 group attach name occurs more than once in the list. Each row must be unique.

**Explanation:** The specified DB2 member for the DB2 group attach name can be used only once.

**System action:** Processing stops.

**User response:** Specify a different DB2 member for the DB2 group attach name.
CCQD525I  The DB2 entries were created.
User response:  No action is required.

CCQD526E  The subsystem_ID DB2 subsystem ID occurs more than once in the list. Each DB2 subsystem ID must be unique.
Explanation:  The specified DB2 subsystem ID can be used only once.
System action:  Processing stops.
User response:  Specify another DB2 subsystem ID.

CCQD527I  DB2 group attach names cannot be created during the copy process.
Explanation:  The ability to create DB2 group attach names is not available during the copy process.
System action:  None.
User response:  Create DB2 group attach names by issuing the CREATE command on the Customizer Workplace panel.

CCQD528E  The metadata_library metadata library is already associated with number DB2 entries. The maximum number of associated DB2 entries for this metadata library is 256.
Explanation:  A metadata library can be associated with a maximum of 256 DB2 entries. The specified metadata library is already associated with 256.
System action:  Processing stops.
User response:  Remove an existing association between the specified metadata library and a DB2 entry, and associate the specified the metadata library with another entry.

CCQD529I  At least one row is required.

CCQD560E  The subsystem_ID DB2 subsystem already exists and is associated with the current product on the Customizer Workplace panel.
Explanation:  The specified DB2 subsystem exists and is associated with the product that you are customizing.
System action:  None.
User response:  Specify another DB2 subsystem.

CCQD561E  The member_name DB2 member for the group_attach_name DB2 group attach name already exists and is associated with the current product on the Customizer Workplace panel.
Explanation:  The specified DB2 member is already associated.
System action:  None.
User response:  Specify a different DB2 member.

CCQD562E  The group_attach_name DB2 group attach name already exists and is associated with the current product on the Customizer Workplace panel.
Explanation:  The specified DB2 group attach name exists and is associated with the product that you are customizing. The subsystem is in the table on the Customizer Workplace panel.
System action:  None.
User response:  Specify another DB2 group attach name.

CCQD563E  A value is required for a DB2 subsystem, a DB2 group attach name, or both before they can be created.
Explanation:  A DB2 subsystem, a DB2 group attach name, or both are not specified so one or both of them cannot be created.
System action:  None.
User response:  Specify a value for the DB2 subsystem, the DB2 group attach name, or both.

CCQD564E  The subsystem_ID DB2 subsystem already exists in the list of DB2 entries and is already associated with the current product.
Explanation:  The specified subsystem is already associated.
System action:  None.
User response:  Specify a different DB2 subsystem.

CCQD565E  The member_name DB2 member for the group_attach_name DB2 group attach name already exists in the list of DB2 entries and is already associated with the current product.
Explanation:  The specified DB2 member is already associated.
System action:  None.
User response:  Specify a different DB2 member.
CCQD567E The group_attach_name DB2 group attach name already exists in the list of DB2 entries and is already associated with the current product.

Explanation: The specified DB2 group attach name is already associated.

System action: None.

User response: Specify another DB2 group attach name.

CCQD568I To customize product_name, at least one DB2 entry must be associated with this product.

Explanation: The specified product requires at least one associated DB2 entry.

System action: None.

User response: To continue the customization process for the specified product, associate one or more DB2 entries with it.

CCQD569I To customize the product_name product configuration, at least one DB2 entry must be associated with this configuration.

Explanation: The configuration for the specified product requires at least one associated DB2 entry.

System action: None.

User response: To continue the customization process for the specified product, associate one or more DB2 entries with the configuration.

CCQD577W The mode_name DB2 mode of the subsystem_ID DB2 subsystem is not supported by the product.

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD578W The mode_name DB2 mode of the member_name DB2 member for the DB2 group is not supported by the product.

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD579W The mode_name DB2 mode of the group_name DB2 group attach name is not supported by the product.

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD580S The subsystem_ID DB2 subsystem was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

Explanation: The copied DB2 subsystem was not associated with the product because the product is associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 subsystem.

CCQD581S The member_name DB2 member for the group_attach_name DB2 group attach name was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

Explanation: The copied DB2 member for the DB2 group attach name was not associated with the product because the product is associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 member.

CCQD582S The group_attach_name DB2 group attach name was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

Explanation: The copied DB2 group attach name was not associated with the product because the product is associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 group attach name.
The member_name DB2 member for the group_attach_name DB2 group attach name is copied to the subsystem_ID DB2 subsystem.

Explanation: The specified DB2 member was copied.

System action: None.
User response: No action is required.

The group_attach_name DB2 group attach name cannot be copied because a DB2 member is required.

Explanation: The specified DB2 group attach name was not copied because a DB2 member was missing.

System action: None.
User response: No action is required.

The current LPAR is LPAR_name, but the data store contains information about the LPAR_name LPAR. You must use the LPAR_name LPAR to customize the product.

Explanation: The LPAR that is stored in the data store data set must be used to customize the product.

System action: Processing stops.
User response: Use the LPAR that is stored in the data store data set.

The level_number DB2 level of the subsystem_name DB2 subsystem is not supported by the product.

Explanation: The product does not support the specified DB2 level.

System action: Processing continues.
User response: Specify a supported level of DB2.

The level_number DB2 level of the member_name DB2 member of the group_name DB2 group is not supported by the product.

Explanation: The product does not support the specified DB2 level.

System action: Processing continues.
User response: Specify a supported level of DB2.

The level_number DB2 level of the group_name DB2 group attach name is not supported by the product.

Explanation: The product does not support the specified DB2 level.

The member_name product customization member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the XML structure of the product customization member is valid, the PL/I
XML parser issued an exception warning code.

**System action:** Processing continues.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQD601S** The member_name product customization member is not valid. The PL/I XML parser issued the following exception error code: code_number.

**Explanation:** While determining if the XML structure of the product customization member is valid, the PL/I XML parser issued an exception error code.

**System action:** Processing continues.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception error code.

---

**CCQD602S** The XML structure of the member_name product customization member is not valid. The element_name element is unknown.

**Explanation:** The data store member contains an unknown element.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

**CCQD603S** The XML structure of the member_name product customization member is not valid. Content is not allowed for the element_name element, but content was found.

**Explanation:** The specified element cannot contain content.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

**CCQD604S** The XML structure of the member_name product customization member is not valid. Content is required for the element_name element, but content was not found.

**Explanation:** The specified element is missing required content.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

**CCQD605S** The XML structure of the member_name product customization member is not valid. The content length for the element_name element exceeds maximum_number characters.

**Explanation:** The specified element contains too many characters.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

**CCQD606S** The XML structure of the member_name product customization member is not valid. The element_name element cannot occur more than maximum_number times.

**Explanation:** The specified element occurs too many times.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

**CCQD607S** The XML structure of the member_name product customization member is not valid. The element_name element must occur at least minimum_number times.

**Explanation:** The specified element does not occur enough times.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

**CCQD608S** The XML structure of the member_name product customization member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

**Explanation:** The specified attribute occurs too many times.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.
CCQD609S  The XML structure of the member_name product customization member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute does not occur enough times.
System action: Processing stops.

CCQD610S  The XML structure of the member_name product customization member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot contain content.
System action: Processing stops.

CCQD611S  The XML structure of the member_name product customization member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute does not contain required content.
System action: Processing stops.

CCQD612S  The XML structure of the member_name product customization member is not valid. The content length for the element_name element exceeds maximum_number characters.

Explanation: The specified element contains too many characters.
System action: Processing stops.

CCQD613S  The XML structure of the member_name product customization member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation: The specified attribute in the data store member is unknown.
System action: Processing stops.

CCQD614S  The content of the member_name product customization member is not valid. The value of the element_name element is not valid. The value is value_name.

Explanation: The specified value is not valid.
System action: Processing stops.

CCQD700W  The member_name DB2 data member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the XML structure of the DB2 data member is valid, the PL/I XML parser issued an exception warning code.
System action: Processing continues.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQD701S  The member_name DB2 data member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the XML structure of the DB2 data member is valid, the PL/I XML parser issued an exception error code.
System action: Processing continues.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception error code.

CCQD750W  The value_number value in the DB2 parameter parameter_name was skipped because only maximum_number values are allowed.

Explanation: The specified value was skipped because
it exceeds the number of allowed values in the DB2 parameter.

**System action:** Processing continues.

**User response:** No action is required. To stop this message from being issued, remove the extra values from the DB2 parameter.

---

**CCQD800W** The member_name LPAR data member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

**Explanation:** While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception warning code.

**System action:** Processing continues.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQD801S** The member_name LPAR data member is not valid. The PL/I XML parser issued the following exception error code: code_number.

**Explanation:** While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception error code.

**System action:** Processing continues.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception error code.

---

**CCQD850W** The value_number value in the LPAR parameter parameter_name was skipped because only maximum_number values are allowed.

**Explanation:** The specified value was skipped because it exceeds the number of allowed values in the LPAR parameter.

**System action:** Processing continues.

**User response:** No action is required. To stop this message from being issued, remove the extra values from the LPAR parameter.

---

**CCQD851I** The subsystem_ID DB2 subsystem is copied to the member_name DB2 member for the group_attach_name DB2 group attach name.

**User response:** No action is required.

---

**CCQD852I** The member_name DB2 member for the group_attach_name DB2 group attach name is copied to the member_name DB2 member for the group_attach_name DB2 group attach name.

**User response:** No action is required.

---

**CCQD854I** The member_name DB2 member for the group_attach_name DB2 group 'attach name is copied to multiple DB2 entries.

**User response:** No action is required.

---

**CCQD900W** The member_name product data member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

**Explanation:** While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception warning code.

**System action:** Processing continues.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQD901S** The member_name product data member is not valid. The PL/I XML parser issued the following exception error code: code_number.

**Explanation:** While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception error code.

**System action:** Processing continues.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception error code.

---

**CCQD950W** The value_number value in the product parameter parameter_name was skipped because only maximum_number values are allowed.

**Explanation:** The specified value was skipped because it exceeds the number of allowed values in the product parameter.

**System action:** Processing continues.

**User response:** No action is required. To stop this message from being issued, remove the extra values from the product parameter.
CCQD960I  The subsystem_ID DB2 subsystem was changed to the member_name DB2 member for the group_attach_name DB2 group attach name.

User response:  No action is required.

CCQD961I  The member_name DB2 member for the group_attach_name DB2 group attach name was changed to the subsystem_ID DB2 subsystem.

User response:  No action is required.

CCQD962I  The member_name DB2 member for the group_attach_name DB2 group attach name was changed to the member_name DB2 member for the group_attach_name DB2 group attach name.

User response:  No action is required.

CCQD963E  The DB2 group attach name cannot be blank when the DB2 subsystem ID is blank.

Explanation:  A DB2 group attach name, DB2 subsystem ID, or both must be specified.

System action:  Processing stops.

User response:  Specify a DB2 group attach name, DB2 subsystem ID, or both.

CCQE000S  The specified message field name or message message_ID was not found.

Explanation:  An error occurred while displaying a message field name or the specified message.

System action:  Processing stops.


CCQE001E  An incorrect trace level was specified. Valid trace levels are 0 - 4.

Explanation:  A wrong trace level was specified. Valid trace levels are 0 - 4.

System action:  Processing stops.

User response:  Specify a valid trace level 0 - 4.

CCQH001W  The specified option option_name is not valid.

Explanation:  The option that was specified is not a valid option on the panel.

System action:  Tools Customizer stops.

CCQH006W  Before you customize a product, verify your user settings.

Explanation:  The user settings must be verified before a product can be customized.

System action:  Tools Customizer stops.

User response:  Verify the user settings.

CCQH007E  Check the user settings. One or more current values are not valid.

Explanation:  One or more of the values in the user settings is not valid.

System action:  Tools Customizer stops.

User response:  Ensure that the specified values for the user settings are valid.

CCQH008W  Before you use Tools Customizer, you must select option 0 to verify your user settings.

Explanation:  The user settings must be changed before a product can be customized.

System action:  Tools Customizer stops.

User response:  Change the user settings.

CCQH009E  You must select option 0 to change your user settings.

Explanation:  User settings must be changed before a product can be customized.

System action:  Tools Customizer stops.

User response:  Change the user settings.

CCQI000W  The XML structure of the member_name DB2 parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:  While determining if the DB2 parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:  Processing continues.

User response:  See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.
**CCQI001S** The XML structure of the *member_name* DB2 parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

**Explanation:** While determining if the DB2 parameter metadata member is valid, the PL/I XML parser issued an exception error code.

**System action:** Processing stops.

**User response:** See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

**CCQI002S** The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *element_name* element is unknown.

**Explanation:** The specified element in the DB2 parameter metadata member is unknown.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251 Contact IBM Software Support.

**CCQI003S** The XML structure of the *member_name* DB2 parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

**Explanation:** The specified element cannot contain content.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251 Contact IBM Software Support.

**CCQI004S** The XML structure of the *member_name* DB2 parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

**Explanation:** The specified element requires content.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251 Contact IBM Software Support.

**CCQI005S** The XML structure of the *member_name* DB2 parameter metadata member is not valid. The content length for the *element_name* element cannot exceed maximum_number characters.

**Explanation:** The specified element contains too many characters.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251 Contact IBM Software Support.

**CCQI006S** The XML structure of the *member_name* DB2 parameter metadata member is not valid. The content length for the *element_name* element must be at least minimum_number characters.

**Explanation:** The specified element does not contain enough characters.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251 Contact IBM Software Support.

**CCQI007S** The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *element_name* element must occur at least minimum_number times.

**Explanation:** The specified element does not occur enough times.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251 Contact IBM Software Support.

**CCQI008S** The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than maximum_number times.

**Explanation:** The specified attribute occurs too many times.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251 Contact IBM Software Support.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI009S</td>
<td>The XML structure of the member_name DB2 parameter metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified attribute did not occur enough times.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI010S</td>
<td>The XML structure of the member_name DB2 parameter metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified attribute cannot have content.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI011S</td>
<td>The XML structure of the member_name DB2 parameter metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified attribute is missing required content.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI012S</td>
<td>The XML structure of the member_name DB2 parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified element contains too many characters.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI013S</td>
<td>The XML structure of the member_name DB2 parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified attribute in the DB2 parameter metadata member is unknown.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI014S</td>
<td>The content of the member_name DB2 parameter metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified value of the element is not a valid value.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI015S</td>
<td>The content of the DB2 parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified value of the attribute is not a valid value.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI016S</td>
<td>The content of the DB2 parameter metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified data type is not a valid data type.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>
CCQI017S The content of the DB2 parameter metadata member is not valid because the data type of the attribute in the element is incorrect. The value of the attribute is value_name.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.


CCQI050S The member DB2 parameter metadata member was not found in the data set.

Explanation: Tools Customizer could not find the specified DB2 parameter metadata member.

System action: Processing stops.


CCQI051S The parameter LPAR parameter in the template template does not have associated metadata in the member LPAR parameter metadata member.

Explanation: The specified template does not contain metadata for an LPAR parameter. The name of the LPAR parameter metadata member, the name of the LPAR parameter, and the name of the template are indicated in the message text.

System action: Processing stops.


CCQI052S The parameter product parameter in the template template does not have associated metadata in the member product parameter metadata member.

Explanation: The specified template does not contain metadata for a product parameter. The name of the product parameter metadata member, the name of the product parameter, and the name of the template are indicated in the message text.

System action: Processing stops.


CCQI053E The following metadata data set was not found: data_set_name.

Explanation: Tools Customizer could not find the specified metadata data set.

System action: Processing stops.

User response: Ensure that the metadata data set is specified correctly. If the problem persists, contact IBM Software Support.

CCQI054E The following metadata data set could not be opened: data_set_name.

Explanation: Tools Customizer could not open the specified metadata data set.

System action: Processing stops.

User response: Ensure the metadata data set was specified correctly.

CCQI055S The product parameter metadata member was not found in the data set.

Explanation: The product parameter metadata member was not found in the specified data set.

System action: Processing stops.


CCQI056S The LPAR parameter metadata member was not found in the data set.

Explanation: Tools Customizer could not find the specified LPAR parameter metadata member.

System action: Processing stops.


CCQI057S The product parameter metadata member was not found in the data set.

Explanation: The product parameter metadata member was not found in the specified data set.

System action: Processing stops.

CCQI058I  Product name does not have any DB2 parameters.
Explanation: DB2 parameters are not required to customize the specified product.
System action: Processing continues.
User response: No action is required.

CCQI059I  Product name does not have any LPAR parameters.
Explanation: LPAR parameters are not required to customize the specified product.
System action: Processing continues.
User response: No action is required.

CCQI060S  The parameter name DB2 parameter in the task description task condition does not have associated metadata in the member name DB2 parameter metadata member.
Explanation: Associated metadata is missing for the specified DB2 parameter in a task.
System action: Processing stops.
User response: See “Gathering diagnostic information” on page 251, Contact IBM Software Support.

CCQI061S  The parameter name LPAR parameter in the task description task and the step description step does not have associated metadata in the member name LPAR parameter metadata member.
Explanation: Associated metadata is missing for the specified LPAR parameter in a task and step.
System action: Processing stops.
User response: See “Gathering diagnostic information” on page 251, Contact IBM Software Support.

CCQI062S  The parameter name product parameter in the task description task condition does not have associated metadata in the member name product parameter metadata member.
Explanation: Associated metadata is missing for the specified product parameter in a task.
System action: Processing stops.
User response: See “Gathering diagnostic information” on page 251, Contact IBM Software Support.

CCQI063S  The parameter name DB2 parameter in the task description task and the step description step does not have associated metadata in the member name DB2 parameter metadata member.
Explanation: Associated metadata is missing for the specified DB2 parameter in a task and step.
System action: Processing stops.
User response: See “Gathering diagnostic information” on page 251, Contact IBM Software Support.

CCQI064S  The parameter name LPAR parameter in the task description task and the step description step does not have associated metadata in the member name LPAR parameter metadata member.
Explanation: Associated metadata is missing for the specified LPAR parameter in a task and step.
System action: Processing stops.
User response: See “Gathering diagnostic information” on page 251, Contact IBM Software Support.

CCQI065S  The parameter name product parameter in the task description task and the step description step does not have associated metadata in the member name parameter metadata member.
Explanation: Associated metadata is missing for the specified parameter in a task and step.
System action: Processing stops.
User response: See “Gathering diagnostic information” on page 251, Contact IBM Software Support.

CCQI066S  The parameter name DB2 parameter in the task description task, step description step, and template name template condition does not have associated metadata in the member name DB2 parameter metadata member.
Explanation: Associated metadata is missing for the specified DB2 parameter in a task, step, and template.
System action: Processing stops.
User response: See “Gathering diagnostic information” on page 251, Contact IBM Software Support.
CCQI067S  The parameter_name LPAR parameter in the task_description task, step_description step, and template_name template condition does not have associated metadata in the member_name LPAR parameter metadata member.

Explanation:  Associated metadata is missing for the specified LPAR parameter in a task, step, and template.

System action:  Processing stops.


CCQI071E  The parameter_name LPAR parameter metadata member is not valid. The default length for the parameter-element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

Explanation:  The specified length cannot be shorter than the default length.

System action:  Processing stops.


CCQI068S  The parameter_name product parameter in the task_description task, step_description step, and template_name template condition does not have associated metadata in the member_name product parameter metadata member.

Explanation:  Associated metadata is missing for the specified product parameter in a task, step, and template.

System action:  Processing stops.


CCQI072E  The parameter_name product parameter metadata member is not valid. The default length for the parameter-element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

Explanation:  The specified length cannot be shorter than the default length.

System action:  Processing stops.


CCQI069S  Product metadata does not support multiple configurations, but the template_name product template contains the parameter_name parameter. Enable multiple configurations support for this product, and try again.

Explanation:  The specified template contains a parameter for multiple configurations, but the product is not enabled to support multiple configurations.

System action:  Processing stops.

User response:  Enable multiple configurations support, and try again.

CCQI073S  The XML structure of the member_name DB2 parameter metadata member is not valid. The following value of the attribute_name attribute in the element_name element already exists: value_name.

Explanation:  The specified value already exists for an attribute.

System action:  Processing stops.

CCQI074S  The XML structure of the member_name LPAR parameter metadata member is not valid. The following value of the attribute_name attribute in the element_name element already exists: value_name.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.


CCQI075S  The XML structure of the member_name product parameter metadata member is not valid. The following value of the attribute_name attribute in the element_name element already exists: value_name.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.


CCQI076S  The XML structure of the member_name DB2 parameter metadata member is not valid. The parameter_name parameter refers to the section-name section. This section was not found in the DB2 parameter metadata member.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.


CCQI077S  The XML structure of the member_name LPAR parameter metadata member is not valid. The parameter_name parameter refers to the section-name section. This section was not found in the LPAR parameter metadata member.

Explanation: The specified parameter refers to a section that is not in the LPAR parameter metadata member.

System action: Processing stops.


CCQI078S  The content of the member_name product parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

Explanation: The specified value for an attribute in the product parameter metadata member is not valid.

System action: Processing stops.


CCQI080S  The content of the member_name DB2 parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

Explanation: The specified value for an attribute in the DB2 parameter metadata member is not valid.

System action: Processing stops.


CCQI081S  The content of the member_name LPAR parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

Explanation: The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action: Processing stops.


CCQI082S  The content of the member_name product parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

Explanation: The specified value for an attribute in the product parameter metadata member is not valid.

System action: Processing stops.

CCQI090S  The product-defined DB2 parameter
parameter_name in the member_name parameter metadata member references the section_ID section ID, but this ID does not exist in either the parameter metadata member or the DB2 parameter metadata member.

Explanation: A section that does not exist in the parameter metadata member or the DB2 parameter metadata member is referenced by the specified DB2 parameter.

System action: Processing stops.


CCQI091S  The product-defined LPAR parameter in the member_name parameter metadata member references the section_ID section ID, but this ID does not exist in either the parameter metadata member or the LPAR parameter metadata member.

Explanation: A section that does not exist in the parameter metadata member or the LPAR parameter metadata member is being referenced by the specified LPAR parameter.

System action: Processing stops.


CCQI092S  The overridden DB2 parameter
parameter_name in the member_name parameter metadata member does not exist in the DB2 parameter metadata member.

Explanation: The specified parameter does not exist.

System action: Processing stops.


CCQI093S  The overridden LPAR parameter
parameter_name in the member_name parameter metadata member does not exist in the LPAR parameter metadata member.

Explanation: The specified parameter does not exist.

System action: Processing stops.


CCQI094S  The CCQ$$PRD product customization parameter metadata member was not found in the data_set_name data set.

Explanation: The specified data set must contain the CCQ$$PRD product customization parameter metadata member

System action: Processing stops.


CCQI100W  The XML structure of the member_name LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQI101S  The XML structure of the member_name LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQI102S  The XML structure of the member_name LPAR parameter metadata member is not valid. The element_name element is unknown.

Explanation: The specified element in the LPAR parameter metadata member is unknown.

System action: Processing stops.

CCQI103S  The XML structure of the member_name LPAR parameter metadata member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.


CCQI104S  The XML structure of the member_name LPAR parameter metadata member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.


CCQI105S  The XML structure of the member_name LPAR parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQI106S  The XML structure of the member_name LPAR parameter metadata member is not valid. The content length for the element_name element must be at least minimum_number characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.


CCQI107S  The XML structure of the member_name LPAR parameter metadata member is not valid. The element_name element must occur at least minimum_number times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.


CCQI108S  The XML structure of the member_name LPAR parameter metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.


CCQI109S  The XML structure of the member_name LPAR parameter metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.


CCQI110S  The XML structure of the member_name LPAR parameter metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI111S</td>
<td>The XML structure of the member_name LPAR parameter metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified attribute is missing required content.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI112S</td>
<td>The XML structure of the member_name LPAR parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified element contains too many characters.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI113S</td>
<td>The XML structure of the member_name LPAR parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified attribute in the LPAR parameter metadata member is unknown.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI114S</td>
<td>The content of the member_name LPAR parameter metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified value for an element in the LPAR parameter metadata member is not valid.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI115S</td>
<td>The content of the member_name LPAR parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified value for an attribute in the LPAR parameter metadata member is not valid.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI116S</td>
<td>The content of the member_name LPAR parameter metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified data type value for an element in the LPAR parameter metadata member is not valid.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI117S</td>
<td>The content of the member_name LPAR parameter metadata member is not valid because the data type of the attribute_name attribute in the element_name element is unknown.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified data type value for an attribute in the LPAR parameter metadata member is not valid.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI120S</td>
<td>The XML structure of the member_name DB2 parameter metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An element contains the specified duplicate value.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See “Gathering diagnostic information” on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>
CCQI121S The XML structure of the member_name LPAR parameter metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.


CCQI122S The XML structure of the member_name discover metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.


CCQI123S The XML structure of the member_name product customization parameter metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.


CCQI124S The XML structure of the member_name product customization parameter metadata member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

CCQI200W The XML structure of the member_name information metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the information metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQI201S The XML structure of the member_name information metadata member is not valid. The element_name element is unknown.

Explanation: The specified element in the information metadata member is unknown.

System action: Processing stops.


CCQI202S The XML structure of the member_name information metadata member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

CCQI204S  The XML structure of the member_name information metadata member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.


CCQI205S  The XML structure of the member_name information metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQI206S  The XML structure of the member_name information metadata member is not valid. The content length for the element_name element must be at least minimum_number characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.


CCQI207S  The XML structure of the member_name information metadata member is not valid. The element_name element must occur at least minimum_number times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.


CCQI208S  The XML structure of the member_name information metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.


CCQI209S  The XML structure of the member_name information metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.


CCQI210S  The XML structure of the member_name information metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.


CCQI211S  The XML structure of the member_name information metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

CCQI212S The XML structure of the member_name information metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQI213S The XML structure of the member_name information metadata member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation: The specified attribute in the information metadata member is unknown.

System action: Processing stops.


CCQI214S The content of the member_name information metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.

Explanation: The specified value for an element in the information metadata member is not valid.

System action: Processing stops.


CCQI215S The content of the member_name information metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: The specified value for an attribute in the information metadata member is not valid.

System action: Processing stops.


CCQI216S The content of the member_name information metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.

Explanation: The specified data type value for an element in the information metadata member is not valid.

System action: Processing stops.


CCQI217S The content of the member_name information metadata member is not valid because the data type of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: The specified data type value for an attribute in the information metadata member is not valid.

System action: Processing stops.


CCQI218S The content of the member_name information metadata member is not valid. The length of the value_name value that of the attribute_name attribute is longer than the value_name value of the attribute_name attribute.

Explanation: The first specified value cannot be longer than the second specified value.

System action: Processing stops.


CCQI219S The content of the member_name information metadata member is not valid. The value_name value of the attribute_name attribute contains the value_name value.

Explanation: The first specified value cannot be longer than the second specified value.

System action: Processing stops.

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI220S</td>
<td>The XML structure of the <strong>member_name</strong> information metadata member is not valid. Content for the <strong>attribute_name</strong> attribute in the <strong>element_name</strong> element exceed <strong>maximum_number</strong> characters.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified attribute contains too many characters.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See <a href="#">“Gathering diagnostic information” on page 251</a>. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI223S</td>
<td>The XML structure of the <strong>member_name</strong> information metadata member is not valid. The value that is specified for the DB2 Level already exists. The value is <strong>value_name</strong>.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified value already exists.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a different DB2 level. If the problem persists, contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI224S</td>
<td>The XML structure of the <strong>member_name</strong> information metadata member is not valid. The value that is specified for the DB2 Mode already exists. The value is <strong>value_name</strong>.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified value already exists.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a different DB2 mode. If the problem persists, contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI250S</td>
<td>The information metadata member was not found in the <strong>data_set_name</strong> data set.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>Tools Customizer could not find the information metadata member in the specified data set.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>If this message was issued on the Specify the Metadata Library (CCQPHLQ) panel, specify the product metadata library. The name of this library is <strong>hlq.SANLDYXIMZDENU</strong>. Do not specify the Tools Customizer metadata library, which is <strong>hlq.SCCQDENU</strong>. If the problem persists, identify the name of the Tools Customizer trace data set and contact IBM Software Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI251E</td>
<td>The <strong>member_name</strong> member was not accessible in the <strong>data_set_name</strong> data set.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified member could not be accessed in the data set.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify the correct metadata library.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI252S</td>
<td>The information metadata member was not found in the <strong>library_name</strong> component metadata library that is part of the <strong>library_name</strong> pack metadata library. The name of the pack is <strong>pack_name</strong>.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified component metadata library does not contain the information metadata member.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify the correct metadata library.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI253E</td>
<td>The <strong>library_name</strong> Tools Customizer metadata library is not current. Update the metadata library on the Tools Customizer Settings panel.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified metadata library is not current.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a current metadata library on the Tools Customizer Settings panel.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI300W</td>
<td>The XML structure of the <strong>member_name</strong> sequence metadata member is not valid. The PL/I XML parser issued the following exception warning code: <strong>code_number</strong>.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception warning code.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing continues.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>See the <a href="#">Enterprise PL/I for z/OS Programming Guide</a> for more information about the exception warning code.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI301S</td>
<td>The XML structure of the <strong>member_name</strong> sequence metadata member is not valid. The PL/I XML parser issued the following exception error code: <strong>code_number</strong>.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception error code.</td>
</tr>
<tr>
<td><strong>System action:</strong></td>
<td>Processing stops.</td>
</tr>
</tbody>
</table>
### CCQI302S  CCQI310S

**User response:** See the [Enterprise PL/I for z/OS Programming Guide](#) for more information about the exception error code, and contact IBM Software Support.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Explanation</th>
<th>System action</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI302S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. The element_name element is unknown.</td>
<td>The specified element in the sequence metadata member is unknown.</td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI303S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. Content is not allowed for the element_name element, but content was found.</td>
<td>The specified element cannot contain content.</td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI304S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. Content is required for the element_name element, but content was not found.</td>
<td>The specified element is missing required content.</td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI305S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. Content length for the element_name element cannot exceed maximum_number characters.</td>
<td>The specified element contains too many characters.</td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI306S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. The element_name element cannot occur more than maximum_number times.</td>
<td>The specified element occurs too many times.</td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI307S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. The element_name element must occur at least minimum_number times.</td>
<td>The specified element does not occur enough times.</td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI308S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.</td>
<td>The specified attribute occurs too many times.</td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI309S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.</td>
<td>The specified attribute does not occur enough times.</td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI310S</td>
<td>The XML structure of the member_name sequence metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.</td>
<td></td>
<td>Processing stops.</td>
<td>See the <a href="#">Gathering diagnostic information</a> on page 251. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>
CCQI311S  CCQI350S


CCQI315S The content of the member_name sequence metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: The specified value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.


CCQI316S The content of the member_name sequence metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.

Explanation: The specified data type value for an element in the sequence metadata member is not valid.

System action: Processing stops.


CCQI317S The content of the member_name sequence metadata member is not valid because the data type of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: The specified data type value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.


CCQI350S The XML structure of the member_name sequence metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: A specified value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.

CCQI351S  The member_name sequence metadata member was not found in the data_set_name metadata data set.

Explanation:  Tools Customizer could not find the specified sequence metadata member in the metadata data set.

System action:  Processing stops.


CCQI352S  The template_name product template was not found in the data_set_name metadata data set.

Explanation:  Tools Customizer could not find the specified product template in the data set.

System action:  Processing stops.


CCQI353S  The sequence metadata member was not found in the data_set_name component data set that is part of the data_set_name pack.

Explanation:  Tools Customizer could not find the sequence metadata member.

System action:  Processing stops.


CCQI360S  The XML structure of the member_name sequence metadata member is not valid. The value of the attribute_name attribute in the element_name element already exists.

Explanation:  The specified attribute contains a value that already exists.

System action:  Processing stops.


CCQI361S  The XML structure of the member_name sequence metadata member is not valid. A relational operator already exists for the condition element on the specified level.

Explanation:  A relational operator already exists for the condition element on the specified level.

System action:  Processing stops.


CCQI362S  The XML structure of the member_name sequence metadata member is not valid. The condition element on the level_type level must contain only one content string or content number element.

Explanation:  Only one content string element or content number element can be contained in the condition element on the specified level.

System action:  Processing stops.


CCQI363S  The XML structure of the member_name sequence metadata member is not valid. The condition element in the element_name element with the attribute_name attribute must contain either the content string element or content number element.

Explanation:  Either the content string element or the content number element must be in the condition element.

System action:  Processing stops.


CCQI400W  The XML structure of the member_name parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:  While determining the parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:  Processing continues.

User response:  See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQI401S  The XML structure of the member_name parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation:  While determining if the parameter
metadata member is valid, the PL/I XML parser issued an exception error code.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQI402S**  The XML structure of the member_name parameter metadata member is not valid. The element_name element is unknown.

**Explanation:** The specified element in the parameter metadata member is unknown.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQI403S**  The XML structure of the member_name parameter metadata member is not valid. Content is not allowed for the element_name element, but content was found.

**Explanation:** The specified element cannot contain content.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQI404S**  The XML structure of the member_name parameter metadata member is not valid. Content is required for the element_name element, but content was not found.

**Explanation:** The specified element requires content.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQI405S**  The XML structure of the member_name parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

**Explanation:** The specified element contains too many characters.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQI406S**  The XML structure of the member_name parameter metadata member is not valid. The content length for the element_name element must be at least minimum_number characters.

**Explanation:** The specified element does not contain enough characters.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQI407S**  The XML structure of the member_name parameter metadata member is not valid. The element_name element must occur at least minimum_number times.

**Explanation:** The specified element does not occur enough times.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQI408S**  The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

**Explanation:** The specified attribute occurs too many times.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQI409S**  The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

**Explanation:** The specified attribute does not occur enough times.

**System action:** Processing stops.

**User response:** See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.
CCQI410S The XML structure of the member_name parameter metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.


CCQI411S The XML structure of the member_name parameter metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.


CCQI412S The XML structure of the member_name parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQI413S The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation: The specified attribute in the parameter metadata member is unknown.

System action: Processing stops.


CCQI414S The content of the member_name parameter metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.

Explanation: The specified value for an element in the parameter metadata member is not valid.

System action: Processing stops.


CCQI415S The content of the member_name parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: The specified value for an attribute in the parameter metadata member is not valid.

System action: Processing stops.


CCQI416S The content of the member_name parameter metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.

Explanation: The specified data type value for an element in the parameter metadata member is not valid.

System action: Processing stops.


CCQI417S The content of the member_name parameter metadata member is not valid because the data type of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: The specified data type value for an attribute in the parameter metadata member is not valid.

System action: Processing stops.

CCQI420S  The XML structure of the member_name parameter metadata member is not valid. The element_name element is unknown for the overridden DB2 parameter.

Explanation:
System action: Processing stops.

CCQI421S  The XML structure of the member_name parameter metadata member is not valid. The element_name element is unknown for the overridden LPAR parameter.

Explanation:
System action: Processing stops.

CCQI422S  The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown for the overridden DB2 parameter.

Explanation:
System action: Processing stops.

CCQI423S  The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown for the overridden LPAR parameter.

Explanation:
System action: Processing stops.

CCQI510W  The data_set_name data store data set does not exist.

Explanation: The specified data store data set does not exist.
System action: Processing continues.
User response: Ensure that the data store data set exists.

CCQI511S  The data_set_name data store data set cannot be opened by using the disposition_type disposition.

Explanation: The specified data store data set could not be opened with the specified disposition.
System action: Processing continues.

CCQI512S  The data_set_name data store data set cannot be opened by using the option-type option.

Explanation: The specified data store data set was unable to be opened with the specified option.
System action: Processing stops.

CCQI600W  The XML structure of the member_name product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the product customization parameter metadata member is valid, the PL/I XML parser issued an exception warning code.
System action: Processing continues.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

CCQI601S  The XML structure of the member_name product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the product
customization parameter metadata member is valid, the
PL/I XML parser issued an exception error code.

**System action:** Processing continues.

**User response:** See the *Enterprise PL/I for z/OS
Programming Guide* for more information about the
warning.

**CCQI602S** The XML structure of the `member_name`
product customization parameter
metadata member is not valid. The
`element_name` element is unknown.

**Explanation:** The specified product customization
parameter metadata member contains an unknown
element.

**System action:** Processing continues.

**CCQI603S** The XML structure of the `member_name`
product customization parameter
metadata member is not valid. Content
is not allowed for the `element_name`
element, but content was found.

**Explanation:** Content was found in an element that
cannot contain content.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic
information" on page 251. Contact IBM Software
Support.

**CCQI604S** The XML structure of the `member_name`
product customization parameter
metadata member is not valid. Content
is required for the `element_name`
element, but content was not found.

**Explanation:** The specified element does not contain
required content.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic
information" on page 251. Contact IBM Software
Support.

**CCQI605S** The XML structure of the `member_name`
product customization parameter
metadata member is not valid. The
`element_name` element 'cannot exceed `maximum_number`
characters.

**Explanation:** The specified element contains too many
characters.

**System action:** Processing stops.

**CCQI606S** The XML structure of the `member_name`
product customization parameter
metadata member is not valid. The
`element_name` element cannot occur more
than `maximum_number` times.

**Explanation:** The specified element occurs too many
times in the product customization parameter metadata
member.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic
information" on page 251. Contact IBM Software
Support.

**CCQI607S** The XML structure of the `member_name`
product customization parameter
metadata member is not valid. The
`element_name` element must occur at least
`minimum_number` times.

**Explanation:** The specified element does not occur
enough times in the product customization parameter metadata
member.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic
information" on page 251. Contact IBM Software
Support.

**CCQI608S** The XML structure of the `member_name`
product customization parameter
metadata member is not valid. The
`attribute_name` attribute in the
`element_name` element cannot occur more
than `maximum_number` times.

**Explanation:** The specified attribute occurs too many
times in the product customization parameter metadata
member.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic
information" on page 251. Contact IBM Software
Support.

**CCQI609S** The XML structure of the `member_name`
product customization parameter
metadata member is not valid. The
`attribute_name` attribute in the
`element_name` element must occur at least
`minimum_number` times.

**Explanation:** The specified attribute does not occur
enough times in the product customization parameter metadata
member.
CCQI610S The XML structure of the member_name product customization parameter metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

CCQI611S The XML structure of the member_name product customization parameter metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute does not contain required content.

System action: Processing stops.

CCQI612S The XML structure of the member_name product customization parameter metadata member is not valid. The content length for the attribute_name attribute in the element_name element cannot exceed maximum_number characters.

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

CCQI613S The XML structure of the member_name product customization parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation: The specified product customization parameter metadata member contains an unknown attribute.

System action: Processing stops.

CCQI614S The XML structure of the member_name product customization parameter metadata member is not valid. The value of the element_name element is not valid. The value is value_name.

Explanation: The specified value of the element is not a valid value.

System action: Processing stops.

CCQI615S The XML structure of the member_name product customization parameter metadata member is not valid. The data type of the element_name element is not valid. The value of the element is value_name.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

CCQI617S The XML structure of the member_name product customization parameter metadata member is not valid. The data type of the attribute_name attribute for the element_name element is not valid. The value is value_name.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.
CCQI650S  The XML structure of the member_name product customization parameter metadata member is not valid. The following value of the attribute already exists: value_name.

Explanation: The specified value for an attribute already exists.

System action: Processing stops.


CCQI651S  The XML structure of the member_name product customization parameter metadata member is not valid. The parameter_name parameter refers to the following section, which was not found in the member_name product customization parameter metadata member: section-name.

Explanation: The specified section is not in the product customization parameter metadata member.

System action: Processing stops.


CCQI652S  The member_name product customization metadata member not valid. The default length for the element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.


CCQI653S  The content of the member_name product customization parameter metadata member is not valid. The value of the attribute attribute_name attribute in the element_name element is not valid. The value of the attribute is value_name.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.


CCQI700W  The XML structure of the member_name solution pack metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

CCQI701S  The XML structure of the member_name solution pack metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the error.

CCQI702S  The XML structure of the member_name solution pack metadata member is not valid. The element_name element is unknown.

Explanation: The specified solution pack metadata member contains an unknown element.

System action: Processing stops.


CCQI703S  The XML structure of the member_name solution pack metadata member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

CCQI704S The XML structure of the member_name solution pack metadata member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.


CCQI705S The XML structure of the member_name solution pack metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQI706S The XML structure of the member_name solution pack metadata member is not valid. The element_name element cannot occur more than maximum_number times.

Explanation: The specified element occurs too many times.

System action: Processing stops.


CCQI707S The XML structure of the member_name solution pack metadata member is not valid. The element_name element must occur at least minimum_number times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.


CCQI708S The XML structure of the member_name solution pack metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.


CCQI709S The XML structure of the member_name solution pack metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.


CCQI710S The XML structure of the member_name solution pack metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.


CCQI711S The XML structure of the member_name solution pack metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute is missing content.

System action: Processing stops.

The XML structure of the member_name solution pack metadata member is not valid. The content length for the attribute_name attribute in the element_name element cannot exceed maximum_number characters.

**Explanation:** The specified attribute contains too many characters.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic information" on page 251. Contact IBM Software Support.

The XML structure of the member_name solution pack metadata member is not valid. The attribute_name attribute in the element_name element is unknown.

**Explanation:** The specified attribute in the solution pack metadata member is unknown.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic information" on page 251. Contact IBM Software Support.

The XML structure of the member_name solution pack metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.

**Explanation:** The specified value of the element is not a valid value.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic information" on page 251. Contact IBM Software Support.

The XML structure of the member_name solution pack metadata member is not valid because the data type of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

**Explanation:** The specified data type is not a valid data type.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic information" on page 251. Contact IBM Software Support.

The XML structure of the member_name solution pack metadata member is not valid because the data type of the element_name element is incorrect. The value of the attribute is value_name.

**Explanation:** The specified data type is not a valid data type.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic information" on page 251. Contact IBM Software Support.

The XML structure of the member_name solution pack metadata member is not valid. The msg element is required for the component_name component that is not customizable.

**Explanation:** The msg element is required for the specified component, which cannot be customized by using Tools Customizer.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic information" on page 251. Contact IBM Software Support.

The solution pack metadata member was not found in the library_name metadata library.

**Explanation:** Tools Customizer could not find the solution pack metadata member in the specified library.

**System action:** Processing stops.

**User response:** See "Gathering diagnostic information" on page 251. Contact IBM Software Support.
CCQI751S  The version in the library_name solution pack metadata library is different than the version in the library_name component metadata library. The name of the pack is pack_name, and the name of the component is component_name.

Explanation:  The version in the solution pack metadata library does not match the version in the component metadata library.

System action:  Processing stops.


CCQI752S  The release in the library_name solution pack metadata library is different than the release in the library_name component metadata library. The name of the pack is pack_name, and the name of the component is component_name.

Explanation:  The release in the solution pack metadata library does not match the release in the component metadata library.

System action:  Processing stops.


CCQI753S  The modification level in the library_name solution pack metadata library is different than the modification level in the library_name component metadata library. The name of the pack is pack_name, and the name of the component is component_name.

Explanation:  The modification level in the solution pack metadata library does not match the modification level in the component metadata library.

System action:  Processing stops.


CCQO000W  The XML structure of the member_name discover parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:  While determining if the discover parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:  Processing continues.

User response:  See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQO001S  The XML structure of the member_name discover parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation:  While determining if the Discover metadata member is valid, the PL/I XML parser issued an exception error code.

System action:  Processing stops.


CCQO002S  The XML structure of the member_name discover parameter metadata member is not valid. The element_name element is unknown.

Explanation:  The specified element in the discover parameter metadata member is unknown.

System action:  Processing stops.


CCQO003S  The XML structure of the member_name discover parameter metadata member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation:  The specified element cannot contain content.

System action:  Processing stops.

CCQO004S  The XML structure of the member_name
discover parameter metadata member is
not valid. Content is required for the
element_name element, but content was
not found.

Explanation: The specified element is missing required
content.

System action: Processing stops.

User response: See “Gathering diagnostic
information” on page 251. Contact IBM Software
Support.

CCQO005S  The XML structure of the member_name
discover parameter metadata member is
not valid. The content length for the
element_name element cannot exceed
maximum_number characters.

Explanation: The specified element contains too many
characters.

System action: Processing stops.

User response: See “Gathering diagnostic
information” on page 251. Contact IBM Software
Support.

CCQO006S  The XML structure of the member_name
discover parameter metadata member is
not valid. The element_name element cannot occur more than
maximum_number times.

Explanation: The specified element occurs too many
times.

System action: Processing stops.

User response: See “Gathering diagnostic
information” on page 251. Contact IBM Software
Support.

CCQO007S  The XML structure of the member_name
discover parameter metadata member is
not valid. The element_name element must occur at least
minimum_number times.

Explanation: The specified element does not occur
enough times.

System action: Processing stops.

User response: See “Gathering diagnostic
information” on page 251. Contact IBM Software
Support.

CCQO008S  The XML structure of the member_name
discover parameter metadata member is
not valid. The attribute_name attribute in
the element_name element cannot occur more than maximum_number times.

Explanation: The specified attribute occurs too many
times.

System action: Processing stops.

User response: See “Gathering diagnostic
information” on page 251. Contact IBM Software
Support.

CCQO009S  The XML structure of the member_name
discover parameter metadata member is
not valid. The attribute_name attribute in
the element_name element must occur at least minimum_number times.

Explanation: The specified attribute does not occur
enough times.

System action: Processing stops.

User response: See “Gathering diagnostic
information” on page 251. Contact IBM Software
Support.

CCQO010S  The XML structure of the member_name
discover parameter metadata member is
not valid. Content is not allowed for the
attribute_name attribute in the
element_name element, but content was
found.

Explanation: The specified attribute cannot contain
content.

System action: Processing stops.

User response: See “Gathering diagnostic
information” on page 251. Contact IBM Software
Support.

CCQO011S  The XML structure of the member_name
discover parameter metadata member is
not valid. Content is required for the
attribute_name attribute in the
element_name element, but content was
not found.

Explanation: The specified attribute requires content.

System action: Processing stops.

User response: See “Gathering diagnostic
information” on page 251. Contact IBM Software
Support.
### CCQO012S
The XML structure of the `member_name` discover parameter metadata member is not valid. The content length for the `attribute_name` attribute in the `element_name` element in the cannot exceed `maximum_number` characters.

**Explanation:** The specified attribute contains too many characters.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

### CCQO013S
The XML structure of the `member_name` discover parameter metadata member is not valid. The `attribute_name` attribute in the `element_name` element is unknown.

**Explanation:** The specified attribute is unknown.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

### CCQO014S
The content of the `member_name` discover parameter metadata member is not valid because the value of the `element_name` element is incorrect. The value is `value_name`.

**Explanation:** A The specified value for an element in the discover parameter metadata member is not valid.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

### CCQO015S
The content of the `member_name` discover parameter metadata member is not valid because the value of the `element_name` element is incorrect. The value is `value_name`.

**Explanation:** The specified value for an attribute in the discover parameter metadata member is not valid.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

### CCQO016S
The content of the `member_name` discover parameter metadata member is not valid because the data type of the `element_name` element is incorrect. The value is `value_name`.

**Explanation:** The specified data type value for an element in the discover parameter metadata member is not valid.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

### CCQO017S
The content of the `member_name` product parameter metadata member is not valid because the data type of the `attribute_name` attribute in the `element_name` element is incorrect. The value is `value_name`.

**Explanation:** The specified data type value for an attribute in the product parameter metadata member is not valid.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

### CCQO050S
The `data_set_name` Discover REXX EXEC data set could not be initialized or was not found.

**Explanation:** Tools Customizer could not find or could not initialize the specified Discover REXX EXEC data set.

**System action:** Processing stops.

**User response:** Ensure that the Discover REXX EXEC is specified correctly.

### CCQO051S
The `data_sharing_group_ID` data sharing group ID cannot contain more than four characters.

**Explanation:** The specified data sharing group ID contains too many characters.

**System action:** Processing continues.

**User response:** Ensure that the specified data sharing group ID does not exceed four characters.

### CCQO052S
The `REXX_EXEC_name` Discover REXX EXEC was not found in the `data_set_name` Discover data set.

**Explanation:** Tools Customizer could not find the Discover REXX EXEC in the specified data set.
System action: Processing stops.
User response: Ensure that the Discover data set was specified correctly.

CCQ053W The **LPAR_name** LPAR name cannot contain more than eight characters.

Explanation: The specified LPAR name contains too many characters.
System action: Processing continues.
User response: Ensure that the specified LPAR name does not exceed eight characters.

CCQ054W The **subsystem_ID** DB2 SSID cannot contain more than four characters. The **record_name** record was not processed.

Explanation: The specified DB2 SSID contains too many characters.
System action: Processing continues.
User response: Ensure that the specified DB2 SSID does not exceed four characters.

CCQ055W The **parameter_name** DB2 group attach name parameter is in the **record_name** Discover record, but a DB2 group attach name was not specified. The record was not processed.

Explanation: The Discover record contains a data sharing group parameter, but a DB2 group attach name was not specified.
System action: Processing continues.
User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQ056W The **parameter_name** DB2 parameter in the **record_name** Discover record did not have a DB2 group attach name or a DB2 SSID. The record was not processed.

Explanation: The Discover record did not have a DB2 group attach name or a DB2 subsystem ID in the DB2 parameter.
System action: Processing continues.
User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQ057W The Discover EXEC could not find the **parameter_name** parameter in the metadata for the product to be customized. The record was not processed.

Explanation: The specified parameter could not be found in the metadata for the product to be customized.
System action: Processing continues.
User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQ058W The **parameter_name** product parameter name in the **record_type** Discover record does not start with **CCQ_LPR_**, **CCQ_DB2_**, or **CCQ_PRD_**. The record was not processed.

Explanation: The parameter in the record does not start with **CCQ_DB2_**, **CCQ_LPAR_**, or **CCQ_PRD_**.
System action: Processing continues.

CCQ059W The **parameter_name** product parameter cannot contain more than 72 characters. The record was not processed.

Explanation: The specified product parameter contains too many characters.
System action: Processing continues.
User response: Ensure that the specified product parameter does not exceed 72 characters.

CCQ060W The **record_name** Discover record from the REXX EXEC output must start with the following record type: **record_type**. The record was not processed.

Explanation: A Discover record from the REXX EXEC output must start with the specified DB2 record type.
System action: Processing continues.

CCQ061I If you do not have a previously customized version of the product, do not run the Discover EXEC. Press END to go to the Customizer Workplace panel.

Explanation: This message is issued when you customize a product for the first time. It prompts you to use the Discover EXEC to discover data from a previous customization of the specified product.
System action: Processing continues.
User response:
Tip: Using the Discover EXEC saves time and reduces errors that can occur when parameters are specified manually. If you want to use the Discover EXEC, specify the required information on the Discover Customized Product Information panel. Otherwise, press End to continue without discovering data from a previous customization of the product.

**CCQO062W** The Discover EXEC could not find the following **parameter_name** parameter in the DB2 metadata. The record was not processed.

Explanation: The specified parameter is missing in the DB2 metadata.

System action: Processing continues.

User response: If this parameter is required, contact IBM Software Support.

**CCQO064W** The Discover-record Discover record did not have a parameter name. The record was not processed.

Explanation: A parameter name was missing in the Discover record.

System action: Processing continues.


**CCQO065W** The value for the **parameter_name** parameter is ignored because it has more than **maximum_number** characters, which is the maximum length that is defined in the metadata. The value is **parameter_value**.

Explanation: The specified value exceeded the maximum allowed length, which was defined in the metadata. Tools Customizer truncated the extra characters.

System action: Processing continues.


**CCQO066W** The **record_name** Discover record from the Discover REXX EXEC output does not have a parameter value. The record was not processed.

Explanation: The Discover record was missing a parameter value from the Discover EXEC output.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

**CCQO067W** The **parameter_name** parameter is defined in the metadata to support one value, but more than one value was found. The last value was used.

Explanation: The definition of the parameter in the metadata supports one value, but more than one value was specified. Only the last value was used.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

**CCQO068W** The value of the **parameter_name** parameter is ignored because the parameter is defined as internal=true. The value is **value_name**.

Explanation: The specified value of the parameter is ignored because it is defined as internal=true.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

**CCQO069W** The Discover EXEC did not find the **parameter_name** parameter in the LPAR metadata. The record was not processed.

Explanation: The specified parameter is missing from the LPAR metadata.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

**CCQO070W** The **record_type** Discover record contains an incorrect delimiter between the Environment section and the Data section. The record was not processed.

Explanation: Tools Customizer found an incorrect delimiter between the Environment section and the Data section.

System action: None.

User response: No action is required.

**CCQO071W** The **member_name** member could not be found in the **data_set_name** Discover data set.

Explanation: Tools Customizer could not find the specified Discover data set.
CCQO072S  CCQP003E

System action: None.
User response: No action is required.

CCQO072S  The member_name discover metadata member was not found in the data_set_name metadata data set.

Explanation: Tools Customizer could not find the specified metadata member in the data set.
System action: Processing stops.

CCQO073E  The member_name discover metadata member is not valid because the default length for the element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

Explanation: The default length for the specified parameter element is longer than the parameter.
System action: Processing continues.
User response: No action is required.

CCQO074S  The content of the member_name discover metadata member is not valid. The value of the attribute_name attribute in the element_name element is not valid. The value of the attribute is value_name.

Explanation: The specified value is not valid.
System action: Processing stops.

CCQO075W  The configuration_ID configuration ID in the record_name Discover record is incorrect. The record was not processed.

Explanation: The specified configuration ID is not correct.
System action: Processing continues.
User response: No action is required.

CCQO076W  The configuration_ID configuration ID cannot contain more than maximum_number characters. The record was not processed.

Explanation: The specified configuration ID contains too many characters.
System action: Processing stops.
User response: Specify a valid value for the DB2 level.

CCQO077S  The discover metadata member was not found in the data_set_name component data set that is part of the data_set_name pack.

Explanation: The discover metadata member was not found in the specified component data set.
System action: Processing stops.

CCQO080I  Product_name does not support the Discover process.

Explanation: The specified product does not support the Discover process.
System action: None.
User response: No action is required.

CCQP000E  The value of the mode_name DB2 mode is not valid for the level_name DB2 level.

Explanation: The specified DB2 mode is not valid for the DB2 level.
System action: Processing stops.
User response: Specify a valid DB2 mode for the DB2 level.

CCQP001E  The value of the mode_name DB2 mode is missing.

Explanation: The specified DB2 mode is not defined.
System action: Processing stops.
User response: Specify a value for the DB2 mode.

CCQP002E  The value of the mode_name DB2 mode is missing.

Explanation: The specified DB2 mode is not defined.
System action: Processing stops.
User response: Specify a value for the DB2 mode.

CCQP003E  The value of the level_name DB2 level is not valid.

Explanation: The specified DB2 level does not have a valid name.
System action: Processing stops.
User response: Specify a valid value for the DB2 level.
CCQP004S The parameter_name parameter does not exist in the CCQ$DB2 DB2 parameter metadata member.

**Explanation:** The CCQ$DB2 DB2 parameter metadata member does not contain the specified parameter.

**System action:** Processing stops.

**User response:** See “Gathering diagnostic information” on page 251. Contact IBM Software Support.

---

CCQP005E The value of the subsystem_ID DB2 SSID is missing.

**Explanation:** The specified DB2 SSID is not defined.

**System action:** Processing stops.

**User response:** Specify a valid value for the DB2 SSID.

---

CCQP006E The value of the group_attach_name DB2 group attach name is missing.

**Explanation:** The specified DB2 group attach name is not defined.

**System action:** Processing stops.

**User response:** Specify a valid DB2 group attach name.

---

CCQQ000E Specify a valid metadata library. Each qualifier of the library must start with an alphabetic character and must be 1-8 alphanumeric characters. The library name must be 1-44 characters.

**Explanation:** The metadata library was not specified in the correct format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

**System action:** Tools Customizer prompts for the correct library name.

**User response:** Specify a library in the correct format. If the message was issued on the Specify the Metadata Library (CCQPHLQ) panel, specify the product metadata library. The name of this library is hlq.SANLDYXIMZDENU.

Do not specify the Tools Customizer metadata library, which is hlq.SCCQDENU.

---

CCQQ001E The data_set_name data set name that was specified for the metadata library was not found.

**Explanation:** The data set does not exist, or the data set name was written in the incorrect format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric.

**System action:** Tools Customizer prompts for a valid data set.

**User response:** Specify a valid data set name.

---

CCQQ002E The data set name that was specified for the library_name metadata library cannot be opened.

**Explanation:** Tools Customizer could not open the data set.

**System action:** Tools Customizer prompts for an available data set.

**User response:** Ensure that the specified data set is available for Tools Customizer to open it.

---

CCQQ003E The data_set_name data set name that was specified for the metadata sample library is not valid. The data set must be in the following format: HLQ.Sxxx.SAMP.

**Explanation:** The specified data set name was not specified in the correct format.

**System action:** None.

**User response:** Specify the data set name in the following format: HLQ.Sxxx.SAMP, where xxx is the three-character prefix for the product.

---

CCQQ004E The data_set_name data set is being used by another user. Try again when the data set is not being used.

**Explanation:** Another user is using the specified data set.

**System action:** None.

**User response:** Ensure that the specified data set is not being used.

---

CCQQ009E The data_set_name data set name that was specified for the metadata library is not valid because the data set is empty.

**Explanation:** The specified data set is empty.

**System action:** Tools Customizer prompts for an available data set.

**User response:** Ensure that the specified data set is available for Tools Customizer to open it.
CCQ011E The *library_name* metadata library for the component that is part of the *library_name* pack was not found in the catalog. The name of the pack is *pack_name*, and the name of the component is *component_name*.

**Explanation:** The specified metadata library is not in the catalog.

**System action:** None.

**User response:** Specify another metadata library.

---

CCQS008E The *data_set_name* Discover data set cannot be opened.

**Explanation:** The specified option could not open the Discover data set.

**System action:** None.

**User response:** Specify a data set to which you have WRITE access.

---

CCQS006W The length of a qualifier for the *data_set_name* customization library data set exceeds 26 characters.

**Explanation:** The qualifier for the customization library data set is too long. The qualifier cannot exceed 26 characters.

**System action:** Processing continues.

**User response:** Specify a qualifier that is 26 characters or less.

---

CCQS007E The discover data set *data_set_name* could not be opened with the *option-type* option.

**Explanation:** The specified option could not open the Discover data set.

**System action:** None.

**User response:** Specify a data set to which you have WRITE access.

---

CCQS008E An error occurred while the *data_set_name* Discover data set was being created.

**Explanation:** While the specified data set was being created, an error occurred.

**System action:** Processing continues.

**User response:** Ensure that you have WRITE authority access to this data set.
CCQS010E  The customization library qualifier is not valid.

Explanation:  The customization library qualifier that was specified is not valid.
System action:  None.
User response:  Specify a valid qualifier for the customization library.

CCQS011E  The group attach option is not valid.

Explanation:  The group attach option that was specified is not valid.
System action:  None.
User response:  Specify a valid option for the group attach option.

CCQS012E  The Tools Customizer metadata library is not valid.

Explanation:  The metadata library that was specified is not a valid data set.
System action:  None.
User response:  Specify a valid data set for the metadata library.

CCQS013E  The Discover data set is not valid.

Explanation:  The Discover data set that was specified is not a valid data set.
System action:  None.
User response:  Specify a valid Discover data set.

CCQS014E  The data store data set is not valid.

Explanation:  The data set that was specified is not a valid data set.
System action:  None.
User response:  Specify a valid data store data set.

CCQS015E  Tools Customizer is already running.

Explanation:  A session of Tools Customizer is already running in your environment. Only one Tools Customizer session is allowed.
System action:  None.
User response:  The trace data set is being used. Free the trace data set, and start Tools Customizer again.

CCQS018E  Information on the first line of the job card exceeds 57 characters.

Explanation:  The first line of the job card can contain only 57 characters. This character limit includes a continuation character.
System action:  Tools Customizer clears the first line of the job card.
User response:  Specify information that does not exceed 57 characters on the first line of the job card.

CCQS019E  The required trace data set, data_set_name, is currently not accessible.

Explanation:  The trace data set must be accessible.
System action:  Processing stops.
User response:  Ensure that the trace data set is accessible.

CCQS020E  An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.

Explanation:  To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.
System action:  None.
User response:  Ensure that ALTER authority for the specified customization library data set is granted.

CCQS021E  The value value_name in the field that contains the cursor position is not valid.

Explanation:  The specified value is not valid.
System action:  None.
User response:  Specify a valid value.

CCQS022E  An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.

Explanation:  To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.
System action:  None.
User response:  Ensure that UPDATE authority for the specified customization library data set is granted.
CCQS023E  An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.

Explanation: To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that UPDATE authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

CCQS024E  An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.

Explanation: To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that ALTER authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

CCQS030E  The following command is not a valid CREATE statement: command_statement.

Explanation: The specified CREATE command statement is invalid because it contains blanks or alphabetic characters.

System action: Processing stops.

User response: Specify a valid CREATE command statement. The correct syntax is CREATE nn, where nn is 1 - 99.

CCQS031E  The following command is not a valid CREATE statement: command_statement. The number that can be specified with the CREATE command is 1 - 99.

Explanation: The specified CREATE command statement is invalid because it contains either 0 or a number greater than 99.

System action: Processing stops.

User response: Specify a valid CREATE command statement. The correct syntax is CREATE nn, where nn is 1 - 99.

CCQT000I  The product configuration ID copied_configuration_ID was successfully copied from configuration_ID.

Explanation: The specified configuration ID was copied.

System action: None.

User response: No action is required.

CCQT001E  The command_name line command was specified more than once, which is not allowed.

Explanation: The specified line command cannot be specified more than one time.

System action: Processing stops.

User response: Specify the line command only once.

CCQT002E  The configuration_ID configuration ID already exists. Specify a different configuration ID.

Explanation: The specified configuration ID exists.

System action: Processing stops.

User response: Ensure that the specified configuration ID is unique.

CCQT003I  The product configuration ID configuration_ID was created.

Explanation: The specified configuration ID was created.

System action: None.

User response: No action is required.

CCQT004I  The product configuration ID configuration_ID was removed.

Explanation: The specified configuration ID was removed.

System action: None.

User response: No action is required.

CCQT005E  The product configuration ID configuration_ID is not valid. The product configuration ID cannot contain a colon (:).

Explanation: The specified configuration ID contains a colon (:), but a colon is not valid.

System action: Processing stops.
Chapter 11. Messages and return codes
**CCQT018E** Specify Y or N, and press Enter.

*Explanation:* A function requires input.

*System action:* Processing stops.

*User response:* To continue, specify Y or N, and press Enter.

---

**CCQT019I** The select configuration_ID configuration process ended.

*Explanation:* The select process for the specified configuration is finished.

*System action:* Processing stops.

*User response:* No action is required.

---

**CCQT020E** The configuration_ID configuration was not created because the data store was not accessible.

*Explanation:* The specified configuration was not created because the data store could not be accessed.

*System action:* Processing stops.

*User response:* Ensure that the data store is accessible and create the configuration again.

---

**CCQT021E** The configuration_ID configuration was not copied because the data store was not accessible.

*Explanation:* The specified configuration was not copied because the data store could not be accessed.

*System action:* Processing stops.

*User response:* Ensure that the data store is accessible and copy the configuration again.

---

**CCQT025I** The configuration_ID configuration was not updated.

*Explanation:* The specified configuration was not updated because the edit process was canceled.

*System action:* Processing stops.

*User response:* No action is required.

---

**CCQT027I** The product configuration was successfully updated.

*Explanation:* The configuration was updated.

*System action:* Processing continue.

*User response:* No action is required.

---

**CCQX001S** Product_name has already been customized by using values from data_set_name data store data set. Switch to the specified data store data set to continue customizing this product.

*Explanation:* The specified product was customized by using values from the specified data store data set.

*System action:* Processing stops.

*User response:* Use the specified data store data set to continue customizing the product.

---

**CCQX002S** component_name has already been customized by using values from data_set_name data store data set. Switch to the specified data store data set to continue customizing this component.

*Explanation:* The specified component was customized by using values from the specified data store data set.

*System action:* Processing stops.

*User response:* Use the specified data store data set to continue customizing the component.

---

**CCQX011I** Product_name was not found.

*Explanation:* The specified product was not found.

*System action:* Processing stops.

*User response:* Specify another product.
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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 Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some 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® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

The term QUNITS used in this publication is a trademark of Innovative Management Solutions, Inc. (IMSI) in the United States or other countries or both.

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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at [www.ibm.com/privacy](http://www.ibm.com/privacy) and the section titled “Cookies, Web Beacons, and Other Technologies” in IBM’s Online Privacy Statement at [www.ibm.com/privacy/details](http://www.ibm.com/privacy/details). Also, see the “IBM Software Products and Software-as-a-Service Privacy Statement” at [www.ibm.com/software/info/product-privacy](http://www.ibm.com/software/info/product-privacy).
Index

Special characters
—26 SQL code returned by DB2 269

A
access path 229, 237
access path hint 166
accessed table information 229
accessibility
overview 18
acctype subparameter 163
active pages 237
ADVISOR parameter 267
ALLINDEX 160
ALLPART parameter 268
alternate index information, including in reports 277
alternative explain tables, specifying 275
analyzing plans, specifying query 275
ANLCNTRL
customizing parameters 101
file 262
ANLCNTRL configuration parameters
BUFF08K 263
BUFF16K 263
BUFF32K 263
BUFFERS 263
COSTING 264
COSTQUN 264
CPUCOST 264
CPU TIME 264
CURRSYM 264
DATASHR 264
DSGROUP 265
DYNAMIC 265
ELAPSED 265
ENGINES 265
ESACOMP 265
ESASORT 265
FASTTBIL 265
IOCALLS 265
IOSCOST 266
LPAEING 266
MONEYIS 266
SETPLAN 266
SUBSYSY 266
SUBVERS 267
TIMCOST 267
ANLCNTRL host parameter file 113, 143
ANLCNTRL parameter
configuring 147
ANLCOST.LOG file 118
ANLCSRD 99
ANLCSRD edit macro in TSO 121
ANLDEBUG 114
ANLI command in TSO 115
ANLIN file 113, 117
ANLINKS parameter 263
ANLOUT file 113
ANLPARM file 145
ANLPARM user parameter file 113, 143
ANLPARM user parameters
ADVISOR 267
BUFFHIT 267
COMMENT 267
CONNECT 267
DBRMKEY 267
DEGREES 269
DELIMIT QUOTE 269
DSPVCOC 270
EEDECALL 270
EXPLAIN 270
HPOLRD 271
KEEPLAN 271
NEWROTA 272
NEWSEEK 272
NEWSTOR 272
NEWXFER 272
NLSCODE 272
OBJECTS 273
PRECISE 273
PROCESS 274
QUALIFY 275
QUERYNO 275
REFRESH 276
REPORTS 276
RETCODE 276
RETPLAN 277
SHOWALT 277
STORAGE 277
USEPLAN 278
VERSION 278
VIADRDA 278
ANLPBRCR stored procedure
CALL parameters 102
using 136
ANLPBRCR stored procedure
CALL parameters 102
using 136
ANLREP file 113
ANLSQPA
QMF intercept 12
SQL parser routine 11
ANLSQSETSQL 247
ANLSTER program 137
ANLSTRPR program 137
ANLWIF
analyzing indexes 124
modifying statistics 124
restoring indexes 130
restoring tables 130
ANLWORK file 113
APPOSTSQL COBOL parameter 269
apply jobs
setting up 130
applying catalog statistics 132
authorization ID 171, 285
authorization table 280
customization (continued)
data sets
customization library 260
data store 260
Discover EXEC library 260
metadata library 260
data store
overview 260
data store data set
specifying 61
DB2 data sharing members
adding 74
associating 74
copying 87
creating 74
DB2 entries 257
adding 74
associating 74
copying 87
creating 74
defining 83
deleting 89
generating jobs for 83
removing 89
selecting 83
specifying 83
unassociating 89
DB2 group attach field
specifying 61
DB2 group attach names
adding 74
associating 74
copying 87
creating 74
DB2 parameters
defining 81
editing 81
DB2 Parameters panel 81
DB2 subsystems
adding 74
associating 74
copying 87
creating 74
defining DB2 parameters 81
defining LPAR parameters 80
defining parameters 77, 83
defining product parameters 77
deleting DB2 entries 89
deleting jobs 68
Discover Customized Product Information panel 72
Discover EXEC
customizing a new version of a product 65, 67
overview 260
retrieving product information automatically 72
Discovered status 257
discovering previous versions 67
discovering product information 72
display options 65
deleting jobs 89
displaying jobs 89
displaying panel text 65
editing LPAR parameters 80
editing parameters 65
editing product parameters 77
Errors in Customization status 257
finding trace data set 251
Finish Product Customization panel 84
first-time 66
customization (continued)
first-time customization 65
generating jobs 83
hiding panel text 65
high-level qualifier 257
Incomplete status 257
job sort order 84
jobs
deleting 90
displaying 89
maintaining 90
renaming 90
sort order 84
submitting 84, 89
LPAR parameters
defining 80
editing 80
LPAR Parameters panel 80
LPARs 90
maintaining jobs 90
master list
adding DB2 entries 74
Associate DB2 Entry for Product panel 74
overview 257
maximizing information panels 65
metadata libraries
specifying 70
metadata library
overview 260
specifying 61
modifying parameters 65
modifying settings 61
multiple instances 61
multiple-LPAR environment 90
Not Required status 257
options 65
panel display options 65
panels
Associate DB2 Entry for Product panel 74
Create a DB2 Entry 74
Customizer Workplace 83
DB2 Parameters 81
Discover Customized Product Information 72
Finish Product Customization 84
LPAR Parameters 80
Product Parameters 77
Specify the Metadata Library 70
parameters
browsing 87
defining 77, 83
viewing 87
preparing to use Tools Customizer 61
product 257
product parameters
changing 68
defining 77
editing 68, 77
modifying 68
Product Parameters panel 77
Ready to Customize status 257
recustomization 65, 68
recustomizing 68
recustomizing a product 65
removing DB2 entries 89
roadmaps 65
customizing for the first time 66
first-time customization 66
customization (continued)
roadmaps (continued)
reustomizing 68
Specify the Metadata Library panel 70
specifying data sets 61
specifying metadata libraries 70
starting Tools Customizer 60
status types
Customized 257
Discovered 257
Errors in Customization 257
Incomplete 257
Not Required 257
Ready to Customize 257
submitting jobs 84
terminology 257
trace data set 251
troubleshooting 251
finding trace data set 251
user job card settings
specifying 61
viewing parameters 87
customization library
overview 260
customization library qualifier
specifying 61
customizing DB2 SQL Performance Analyzer 143
customizing settings 61

D
data compression, specifying feature 265
data definition 237
data pages, limiting number 278
data sharing
specifying number of members 265
specifying percentage 264
data store
overview 260
data store data set
specifying 61
DATASHR parameter 264
DB2 Administration Tool interface 98
DB2 call attachment facility 12
DB2 catalog 11
DB2 catalog access 5
DB2 catalog tables 151
DB2 Explain 11
DB2 group attach field
specifying 61
DB2 release 237
DB2 SQL Performance Analyzer
generic plan tables
introduction 5
performance features 4
stored procedure
calling 102
customizing parameters 101
operational requirements 27
specifying target host configuration file 100
DB2 SQL Performance Analyzer panels
change user parameters panel 118
main control panel 117
welcome panel 115
DB2 SQL Performance Analyzer stored procedure 6, 137
using 136
DB2 subsystem name, specifying 266

DBRM information 229
dbrm subparameter 164
DBRMKEY parameter 113, 267
defaults data set 172
definition of data 237
DEGREES parameter 269
delete info in Enhanced Explain report 203
deletion of plan records, preventing 271
DELIMIT QUOTE parameter 269
DETail 160
detail trace report 220
detail Trace report 113, 206
diagnostic information
gathering 247, 251
Discover EXEC
overview 260
display options 65
displaying panel text 65
documentation
accessing 17
sending feedback 17
documentation changes 3
DRDA 6
DRDA protocol 142
DSN_STATEMNT_TABLE use
PRECISE parameter 273
service units 273
DSN3@ATH exit
secondary authorization IDs 12
DSN3@SGN exit
secondary authorization IDs 12
dSNSPAS startup procedure 100
DSPWCOC 7
dynamic explain 13
DYNAMIC parameter 265

E
Easy Explain 13, 153
data source 140
tabular report reference 240
Easy Explain parameters 267
Easy Explain report 229, 237
EEECALL parameter 270
ELAPSED parameter 265
engines
specifying number dedicated to partition 266
specifying number in processor 265
ENGINES parameter 265
Enhanced Explain report
description 179
in batch 113
in TSO 117
ESACOMP parameter 265
ESASORT parameter 265
EXPLAIN parameter 270
EXPLAIN parameters
DSPWCOC 270
EEECALL 141, 270
EEEPATH 141
EXPLAIN 141, 270
QUERYNO 141, 275
STOPNO 141
STOPQNO 277
using 141

422 User's Guide
metadata library
overview 260
specifying 61
migrating DB2 SQL Performance Analyzer 106
migrating statistics 130
migration parameters
applying 132
specifying 131
modifying settings 61
modifying statistics 130
MONEYIS parameter 266
multiple DBRM support 113
multiple DBRMs, specifying search pattern 267
multiple index scan info in Enhanced Explain report 192

N
nested loop join info in Enhanced Explain report 195
NEWROTA parameter 272
NEWSEEK parameter 272
NEWSTOR parameter 272
NEWXFER parameter 272
NLS parameter 170
NLSCODE parameter 272
NOCatalog 160
NOKeys 160
non-matching index scan 163
NONIXPG 272
NONONIX parameter 273
NOSTATS parameter 273
NOTSCAN parameter 273

O
OBJECTS parameter 273
OPTIMIZE FOR n ROWS clause 211
options 65
output report panel 118
owner subparameter 153, 168

P
package report 232
packages subparameter 168
pages with rows 237
panel display options 65
panels
Copy DB2 Entries 87
parallel processing 269
parameter configuration
using ISPF 145
parameters
showing 118
TOPA 141
viewing 118
PARMS field 163, 170
Path Checker input to DB2 SQL Performance Analyzer 220
path length estimates in Detail Trace report 215
path table 284
performance features 4
performance notes and recommendations, specifying
activation 267
PKG 151
PLAN 151
plan history report 237
plan table 282
PLAN_TABLE 151, 152, 285
PLAN_TABLE authorization ID 171
PLAN_TABLE specification 278
planinfo subparameter 169
PLN 151
PRECISE parameter 148, 273
predicate analysis report 226
predicate analysis report, enabling 148
PREDPCT parameter 274
preparing to use Tools Customizer 61
problems
  diagnostic information about 247, 251
PROCESS parameter 274
processing
  DBCS 272
  mixed data names 272
processing remotely 141
processing step info in Detail Trace report 211
PROMPTED query 151
Q
QBE query 151
QLIMIT file example 223
QLIMIT file record format 223
QLIMIT report
  in batch 113, 223
  in TSO 223
QLIMSQL 275
QMF 16
  batch processing 136
  canceling queries 135
  governor exit 12, 134
  intercept exit 134
  interface
    creating 94
    described 5
  PRECISE parameter in QMF 148
  literals 151
  parameters 151
  system tables 151
  QMF governor exit 134
  QNO 151
  QTRACE file 113
QUALIFY parameter 275
query CPU time 7
Query Monitor interface 93
QUERY number 176
QUERYNO parameter 275
QUNITS 177
quotation marks and COBOL programs 269
QUOTE parameter 269
QUOTESQL COBOL parameter 269
R
random matching index scan info in Enhanced Explain report 190
reader comment form 17
REFRESH parameter 276
relative cost, QUNITS 177
remote
  package 236
  report 236
remote operation 6
remote processing 141
report append option 118
report overlay option 118
reports menu 133
REPORTS parameter 276
reports, defining level of reporting 276
reports, specifying type 276
resource consumption estimates 176
resource estimator 12
Resource Limit Facility (RLF) 6
restoring indexes 130
restoring tables 130
RETCODE parameter 276
RETPLAN parameter 277
return codes 289
roadmaps
  customizing for the first time 66
  first-time customization 66
rotational delay, specifying 272
routing processing 278
row filtering, setting percentage 274
RUN statement 170
run time 162
S
SANLDATA and SANLPARM, difference 149
screen readers and magnifiers 18
scrollable fields 116
secondary authorization IDs 12
secondary SQL ID 285
SECONDS OF ELAPSED TIME 176
seek time, specifying 272
service information 17
SET command 163
SETPLAN parameter 266
setting up apply jobs 130
setting up collect jobs 130
show parameters 118
SHOWALT parameter 277
SMF accounting 7
sort info in Enhanced Explain report 201
source input notification for SQL statements 270
source type 237
specification of
  Comment statement 153
  Level statement 153
  Package statement 153
  Plan statement 153
  QMF statement 153
  QNO statement 153
  SET statement 153
  SQL statement 153
specifying (continued)
  DB2 connection type for query 267
  DB2 subsystem name 266
  DB2 version for query 278
  DELIMIT QUOTE 269
dynamic statement cache 265
EEXECALL 270
EXPLAIN 270
hardware-assisted data compression 265
hardware-assisted sort 265
high-level qualifier for query DB2 objects 275
hiperpool hit ratio for query 271
host DB2 subsystem 267
host parameter file in batch 113
input from Path Checker 113
KEEPPLAN 271
materialized query tables for access path selection 276
monetary cost limit 264
monetary cost of CPU time 264
monetary cost of I/Os 266
multiple DBRMs 267
national currency 266
national currency symbol 264
NLSCODE 272
number of 16K buffers 263
number of 32K buffers 263
number of 4K buffers 263
number of 8K buffers 263
number of data sharing group members 265
number of engines for DB2 266
number of engines in processor 265
parallel processing for query 269
percent of DB2 work in data sharing 264
physical I/O limit 265
plan table 278
QUERYNO 275
report file disposition TSO 118
report files in batch 113
report files in TSO 117
SQL input
  in batch 113
  in TSO 117
SQL PA drop overheads 263
SQL PA use Class 1 CPU 263
SQL parsing only 263
STOPQNO 277
temporary files in batch 113
time 273
time limit 265
tips for new SQL coders 267
type of reports for query 276
use of COMMENT 264
use of USEPLAN 266
user parameter file in batch 113
user parameters in TSO 118
version recognition 274
work file for multiple DBRM support in batch 113
specifying data sets 61
SQL Advisor 7
SQL input
  in batch 113
  in QMF 134
  in TSO 117
SQL PA 173
codes 289
generic plan tables
  resetting the REGISTRY 247
SQL PA (continued)
messages 289
predicate analysis report 148
return codes 289
SQL PA panels
output report panel 118
show parameters panel 118
SQL PA parameters
ADVISOR parameter 267
ALLPART 268
ANLKEYS 263
BUFF08K 263
BUFF16K 263
BUFF32K 263
BUFFERS 265
BUFFHIT 267
COMMENT 267
CONNECT 267
COSTING 264
COSTQUN 264
CPUCOST 264
CPUTIME 264
CURRSYM 264
DATASHR 264
DBRMKEY 267
DEGREES 269
DELIMIT QUOTE 269
DSGROUP 265
DYNAMIC 265
EEECALL 270
ELAPSED 265
ENGINES 265
ESACOMP 265
ESASORT 265
EXPLAIN 270
FASTTBL 265
HPOOLRD 271
influence on cost estimates 143
INLISTS 271
IOCALLS 265
IOSCOST 266
ISCANPG 271
IXUPDAT 271
JOINTAB 271
KEEPPLAN 271
listed in Detail Trace report 207
listed in Enhanced Explain report 179
LPARENG 266
MATCOLS 271
MESSAGE 272
MONEYIS 266
NEWROTA 272
NEWSEEK 272
NEWSTOR 272
NEWXFER 272
NLSCODE 272
NONIXPG 272
NONONIX 273
NOSTATS 273
NOTSCAN 273
OBJECTS 273
PRECISE 273
PREDPCT 274
PROCESS 274
QUALIFY 275
QUERYNO 275
REFRESH 276
SQL PA parameters (continued)
REPORTS 276
RETCODE 276
RETPLAN 277
SETPLAN 266
SHOWALT 277
STOPQNO 277
STORAGE 277
SUBSyst 266
SUBVERS 267
TIMCOST 267
TSCANPG 278
TURNOFF 278
USEPLAN 278
VERSION 278
VIADRDA 278
SQL parser routine 11
SQL Performance Analyzer 173
SQL, in batch JCL 111
SQLsummary 160
statistics modification 124
STEPLIB user parameter file 113
STOPQNO parameter 277
storage device, defining 272
STORAGE parameter 277
stored procedure 16
  call parameters 102
  calling 102
  customizing parameters 101
  operational requirements 27
  specifying target host configuration file 100
  using ANLPRCR 136
  using ANLPRER 136
  using SQL PA 136
stored procedure customization 100
subparameter
  acctype 163
  dbrm 164
  first 164
  force 165
  format 165
  gen 165
  hostvar 166
  index 153, 167
  last 164
  level 167
  location 168
  owner 153, 168
  packages 168
  planinfo 169
  table 169
  TOPA 170
  SUBSyst parameter 266
  SUBVERS parameter 267
  SUMMARY 161
  summary of changes 3
  summary report 138, 151, 232
  support
    required information 247, 251
    support information 17
  syntax diagrams
    how to read vii
  SYSPRINT file 113
  system parameters, viewing 118
  system tables
    DB2 151
    QMF 151
system thresholds, viewing 118

T

threshold control parameters (continued)
NOMIXPG 272
NOMIXONE 273
NOTSCANS 273
PREDPCT 274
TSCANPG 278
TURNOFF 278

time limit, specifying 265
TINO 161
TIVES 161

Tools Customizer
associated list
adding DB2 entries 74
overview 257
associating DB2 entries 74
browsing parameters 87
changing display options 65
component 257
Copy DB2 Entries panel 87
copying DB2 entries 87
Create a DB2 Entry panel 74
creating DB2 entries 74
customization jobs
deleting 90
displaying 89
generating 83
maintaining 90
renaming 90
sort sequence 84
submitting 84, 89
customization library
deleting jobs 90
maintaining 90
recustomizing 90
renaming jobs 90
customization library qualifier
specifying 61
Customized status 257
Customizer Workplace panel 83
customizing a new version of a product 65, 67
customizing a product for the first time 65, 66
data sets
customization library 260
data store 260
Discover EXEC library 260
metadata library 260
data store data set
specifying 61
DB2 data sharing members
adding 74
associating 74
copying 87
creating 74
DB2 entries 257
adding 74
associating 74
copying 87
creating 74

tables
EEEAUTH 280
EEEDBRM 281
EEEHINT 282
EEEPATH 284
EEEPPLAN 282
EEEWORK 280
PLAN_TABLE contents 285

Tables and index names 151
tables per table space 237
tables, limiting number to join 271

TABular 161
tabular report reference

Easy Explain 240
target host configuration parameters
ANLCNTL parameter file 113
SUBVERS 267
target host parameter file 113
target host parameters
ANLKEYS 263
BUFF08K 263
BUFF16K 263
BUFF32K 263
BUFFERS 263
COMMENT 264
COSTING 264
COSTQUN 264
CPUCOST 264
CPUSUM 264
CURRSYM 264
DATASHR 264
dscroup 265
DYNAMIC 265
ELAPSED 265
ENGINES 265
ESACOMP 265
ESASORT 265
FASTTXT 265
IOCALLS 265
IOSCOST 266
LPARENG 266
MONEYIS 266
SETPLAN 266
SUBSYS 266
TIMCOST 267
target host system, selecting 147
technotes 17
temporary files in batch
ANLOUT file 113
test programs
ANLSTPR 137
threshold control parameters
ALLPART 268
LNOISTS 271
ISCANPG 271
IXUPDAT 271
JOINTAB 271
MARCH 271
MESSAGE 272

Index 427