

User's Guide

!DB®/EXPLAIN for DB2

Version 500

Document Number TE54-5842-3

June 2000

Candle Corporation
201 North Douglas Street
El Segundo, California 90245

Registered trademarks and service marks of Candle Corporation: AF/OPERATOR, AF/PERFORMER, AF/REMOTE, Availability Command Center, Candle Command Center, Candle Electronic Customer Support, Candle Logo, Candle Management Server, Candle Management Workstation, Candle Technologies, CL/CONFERENCE, CL/SUPERSESSION, CT, CT/Data Server, CT/DS, DB Logo, DB/QUICKCHANGE, DELTAMON, ETEWatch, IntelliWatch, MQSecure, MQView, OMEGACENTER, OMEGAMON, OMEGAMON/e, OMEGAMON II, OMEGAMON Monitoring Agent, OMEGAVIEW, OMEGAVIEW II, Solutions for Networked Businesses, and Transplex.

Trademarks and service marks of Candle Corporation: Alert Adapter, Alert Adapter Plus, Alert Emitter, AMS, Amsys, AUTOMATED FACILITIES, Availability Management Systems, Candle Business Partner Logo, Candle Direct Logo, CandleLight, CandleNet, CandleNet 2000, CandleNet Portal, CCC, CECS, CICAT, CL/ENGINE, CL/GATEWAY, CL/TECHNOLOGY, CMS, CMW, Command & Control, CommandWatch, Connect-Two, CSA ANALYZER, CT/ALS, CT/Application Logic Services, CT/DCS, CT/Distributed Computing Services, CT/Engine, CT/Implementation Services, CT/IX, CT/Workbench, CT/Workstation Server, CT/WS, DB/DASD, DB/EXPLAIN, DB/MIGRATOR, DB/QUICKCOMPARE, DB/SMU, DB/Tools, DB/WORKBENCH, Design Network, DEXAN, eBA*ServiceMonitor, End-to-End, Enterprise Candle Command Center, Enterprise Candle Management Workstation, EPILOG, ERPNet, ESRA, HostBridge, IntelliWatch Pinnacle, Lava Console, Messaging Mastered, MQADMIN, MQEdit, MQEXPERT, MQMON, NBX, OMA, OMC Gateway, OMC Status Manager, OMEGACENTER Bridge, OMEGACENTER Gateway, OMEGACENTER Status Manager, OMEGAMON Management Center, OSM, PC COMPANION, Performance Pac, PowerQ, PQConfiguration, PQEdit, PQScope, Response Time Network, Roma, Roma Broker, Roma BSP, Roma Connector, Roma Developer, Roma FS/A, Roma FS/Access, Roma Network, Roma Systems Manager, Roma Workflow Access, Roma WF/Access, RTA, RTN, SentinelManager, Solutions for Networked Applications, Status Monitor, Tracer, Unified Directory Services and Volcano.

Trademarks and registered trademarks of other companies: AIX, DB2, and MQSeries are registered trademarks of International Business Machines Corporation. SAP is a registered trademark and R/3 is a trademark of SAP AG. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Ltd. HU-UX is a trademark of Hewlett-Packard Company. SunOS is a trademark of Sun Microsystems, Inc.

All other company and product names used herein are trademarks or registered trademarks of their respective companies.

ProtoView Development Corp. - May contain DataTable Version 3.0 Copyright 1989—1996 by ProtoView Development Corp. and distributed under license by Candle Corporation.

Copyright © 1991, 2000, Candle Corporation, a California corporation. All rights reserved. International rights secured.

Threaded Environment for AS/400, Patent No. 5,504,898; Data Server with Data Probes Employing Predicate Tests in Rule Statements (Event Driven Sampling), Patent No. 5,615,359; MVS/ESA Message Transport System Using the XCF Coupling Facility, Patent No. 5,754,856; Intelligent Remote Agent for Computer Performance Monitoring, Patent No. 5,781,703; Data Server with Event Driven Sampling, Patent No. 5,809,238; Threaded Environment for Computer Systems Without Native Threading Support, Patent No. 5,835,763; Object Procedure Messaging Facility, Patent No. 5,848,234; Communications on a Network, Patent Pending; End-to-End Response Time Measurement for Computer Programs, Patent No. 5,999,705; Improved Message Queuing Based Network Computing Architecture, Patent Pending; User Interface for System Management Applications, Patent Pending.

NOTICE: This documentation is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions set forth in the applicable license agreement and/or the applicable government rights clause.

This documentation contains confidential, proprietary information of Candle Corporation that is licensed for your internal use only. Any unauthorized use, duplication, or disclosure is unlawful.

Read This First	11
Preface	11
Contents of this Book	12
Adobe Portable Document Format	14
Documentation Conventions	15
Documentation Set	18
What's New	21
Version 500 Changes	21
Getting Started	23
Chapter 1. Introducing !DB/EXPLAIN	25
Introduction	25
Chapter contents	25
Summary of !DB/EXPLAIN's Major Functions	26
Benefits of !DB/EXPLAIN's Extract Technology	27
Benefits for Database Administration	28
Benefits for Application Development	29
Support for Improved System Performance	31
Chapter 2. Introducing !DB/EXPLAIN Information	33
Introduction	33
Chapter Contents	33
Structure of this Guide	34
Using !DB/EXPLAIN Online Help	36
Elements of a !DB/EXPLAIN Help Panel	37
Using Associated Publications	40
Chapter 3. !DB/EXPLAIN Basics	41
Introduction	41
Chapter contents	41
What is an EXPLAIN?	42
Setting Up Your EXPLAIN Output	44
Elements of Your EXPLAIN Output	46
The BIND Section	47
The Statements Section	49
Running an EXPLAIN	54
EXPLAINing SQL Statements	56
Displaying !DB/EXPLAIN Information	57
Comparing Information About Objects	59
Reviewing Exception Conditions	61
Changing Values and Playing What-if	62

Using !DB/EXPLAIN Facilities to Generate JCL for Common Tasks	63
Binding or Rebinding Objects	64
Other Basic Functions You Can Perform	66
Introducing Fundamentals	67
Chapter 4. Accessing and Exiting !DB/EXPLAIN	69
Introduction	69
Chapter contents	69
Accessing !DB/EXPLAIN	70
Access from the !DB/Tools Product Selection Menu	71
Access by Invoking the !DB/EXPLAIN CLIST	73
Access from an Edit Macro	77
Access Using the Edit Macro KTEXPL	78
Initiating an EXPLAIN using KTEXPL	81
Access Using the Edit Macro KTEXPLB	86
Access Using the EXEC KTEQMF	87
Access Using the EXEC KTEON	90
Access from EXEC KTEXPLA	92
Access from EXEC KTEXPLBA	95
Access from EXEC KTEXPLD	98
Determining Whether to Display Statements	101
Access from Candle's OMEGAMON II for DB2 (First Access)	104
Accessing !DB/EXPLAIN Batch Facilities	105
Exiting !DB/EXPLAIN	106
Chapter 5. Accessing !DB/EXPLAIN Functions	107
Introduction	107
Chapter contents	107
Access from the Primary Menu	108
Using the Administration Menu	110
Using the JCL Generation Menu	112
Access from the Sessions Menu	114
Methods for Working with Sessions	116
Restrictions Applicable to Working with Sessions	118
Using the Functions of the !DB/EXPLAIN Online Menu	119
Processing Multiple Datasets on the Online Menu	122
Processing Multiple Members on the Online Menu	123
Activities You Can Perform from a Typical !DB/EXPLAIN Panel	125
Chapter 6. Using Commands	127
Introduction	127
Chapter contents	127
Determining What Commands are Available	128
Viewing Available !DB/EXPLAIN Commands	129

Method for Using Commands	132
Commands for Working with the User PDS	134
Chapter 7. Using Selects	135
Introduction	135
Chapter contents	135
Information about Selects	136
Determining what Selects are Available	137
Fast Path for Using Selects on Object List Panels	138
Methods to Recover from Errors in Processing	140
Chapter 8. Using Sorts	141
Introduction	141
Chapter contents	141
Defining the Default Sort	142
Accessing the Sorts Menu	143
Invoking a Sort	144
Returning a !DB/EXPLAIN Panel to Its Original Order ..	146
Chapter 9. Using Filters	147
Introduction	147
Chapter contents	147
Introduction to Filtering	148
Defining Default Filter Types	150
Invoking Filtering	151
Subsetting the Objects to Display	155
Resetting Filters	158
Effects of Filtering	159
Setting Up a Filtering Macro	160
Using a Filtering Macro	161
Chapter 10. Viewing Alternate Formats	163
Introduction	163
Chapter contents	163
What is an Alternate Format?	164
Defining the Default Format	165
Accessing the Formats Menu	167
Viewing a Specific Format	169
Chapter 11. Using ISPF Facilities within !DB/EXPLAIN	171
Introduction	171
Chapter contents	171
Performing Some Common Operations under ISPF	172
Returning to the !DB/EXPLAIN Primary Menu	175
Controlling the Operation of Scrolling	176
Using the HELP Command to Display Information	179

Using !DB/EXPLAIN System Facilities	181
Chapter 12. Using the Macro Facility	183
Introduction	183
Chapter contents	183
Background about Macros	184
Creating Macros	185
Invoking a Macro	186
Results of Executing a Macro	187
Example Macros Shipped with !DB/EXPLAIN	188
Chapter 13. Using !DB/EXPLAIN with OMEGAMON II for DB2 ..	191
Introduction	191
Chapter contents	191
Background about OMEGAMON II for DB2	192
Using OMEGAMON II and !DB/EXPLAIN Complementary Functions	193
Using the OMEGAMON II for DB2 Callable Interface ..	195
Moving Interactively between !DB/EXPLAIN and OMEGAMON II	209
Using OMEGAMON II's Copy-and-Paste Facility	213
Accessing OMEGAMON II Trace Information from !DB/EXPLAIN	215
Chapter 14. Support for Distributed DB2	217
Introduction	217
Chapter contents	217
Prerequisites for Using !DB/EXPLAIN Distributed Support	218
Remote Functions Available with !DB/EXPLAIN	220
Commands and Selects to Perform Remote EXPLAINS ..	221
Performing Remote EXPLAINS to Support Application Development	224
Commands to Gather or Update Remote Catalog Statistics	226
Commands to Perform Remote BINDS/REBINDs for Packages	227
Specifying the Remote DB2 Subsystem	228
Working With a Remote EXPLAIN	231
Some Usage Scenarios	233
Performing Remote Functions in Batch	235
Performing !DB/EXPLAIN Tasks	237
Chapter 15. Example Tasks	239
Introduction	239
Chapter contents	239
Accessing !DB/EXPLAIN Using the KTEON EXEC	240

Accessing !DB/EXPLAIN Using the KTEON EXEC to Generate JCL	244
Execute an SQL Statement and Review its Results	252
Analyze a High-Cost Package SQL Query Using the Costs Panel	256
Print the Results of a Cost Analysis	264
Execute a “Merge Since” Extract	266
EXPLAIN SQL Statements in a Source Program and Review the Results	267
Set Up General Housekeeping Options to Customize !DB/EXPLAIN	275
Convert a Plan DBRM to a Package Using the Z Select	281
Convert Multiple Plan DBRMs to Packages Using the BINDCNV Command	285
Convert DBRMs in Multiple Plans to Packages Using the BINDCNV Command	290
Change All Plans with an Isolation Level of RR (Repeatable Read) to CS (Cursor Stability)	295
Customizing !DB/EXPLAIN	301
Chapter 16. Controlling the Format of the EXPLAIN Display	303
Introduction	303
Chapter contents	303
Information Available in an EXPLAIN	304
Controlling What Information is Generated	306
Customizing Your EXPLAIN Output	311
Initial EXPLAIN Format	312
Changing the Initial Format During a Session or Batch Job	313
Changing the Format 0 Display	316
Chapter 17. Controlling the Destination of !DB/EXPLAIN Output	319
Introduction	319
Chapter contents	319
Using Output Options	320
Using the Count Variable on Commands and Selects	322
Using the MOD Option on Commands and Selects	325
Viewing Profile Datasets	326
Setting DB/Tools Profile Datasets	328
The Extract Program	331
Chapter 18. The !DB/EXPLAIN Extract Program and the Extract ID	333
Introduction	333
Chapter contents	333
Definition and Features of an Extract	334

Background about the Extract ID	336
Viewing Available Extract IDs	338
Selecting an Extract ID to Work With	341
Defining the Extract ID	342
How Many Extract IDs Should You Define?	344
Customizing Extract IDs in Your System Profile dataset ..	346
Extract Masking	348
Chapter 19. Running the Extract	351
Introduction	351
Chapter contents	351
Planning for Extract Runs	352
Executing the Extract	354
Purpose of Your Extract Run	359
Function Options for Normal Extract Runs	361
Function Options for Normal Purge Runs	366
Normal Extract and Purge Summary	367
Input Syntax	370
Extract Input Examples	373
Purge Input Examples	375
Other Types of Extract Runs	376
Debugging Your Extract	378
Using Extract History	380
Chapter 20. Disk Storage for the Extract	381
Introduction	381
Chapter contents	381
Estimating Disk Storage Required by an Extract	382
Estimating Space for Non-VSAM Files	386
Planning for VSAM Clusters	388
VSAM Fragmentation and REORG Considerations	390
Manual Reorganization	392
Automatic Reorganization	394
Recovering Your VSAM Clusters	395
Restarting after a Failure	396
Using Batch Facilities	399
Chapter 21. Using !DB/EXPLAIN in Batch	401
Introduction	401
Chapter contents	401
Planning for Running Batch Jobs	402
Determining Which Batch Job to Use	403
Example JCL and Control Members	404
Using the JCL? Command to Create a Batch Job	407
Using the Distributed JCL to Create a Batch Job	410
Creating Your Own Batch Job	411
How to Submit a Batch Job	412

Options for Batch Jobs	413
Changing the Value of Batch Options	415
Building Profile Variable Input for Batch Processing Using PROFOUTE	417
Additional Considerations for Batch Processing	418
Chapter 22. Batch Processing for Application Development	419
Introduction	419
Chapter contents	419
JCL for Batch Application Development	420
Modifying the JCL—Common Parameters	423
Modifying the JCL—Parameters Specific to KTEBSQLS	425
Modifying the JCL—Parameters Specific to KTEBDBR2	427
Modifying the JCL—Parameters Specific to KTEBDBRM	428
Input to Your Batch Job	429
Putting It Together—Flow of Your Batch Job	432
Bypassing the Statements Display in Batch	437
Chapter 23. Batch Processing for Database Administration	439
Introduction	439
Chapter contents	439
JCL for Batch Database Administration and Management	440
Input to Your Batch Job	442
A Comprehensive Example	452
Problem Resolution	455
Chapter 24. Resolving Problems	457
Introduction	457
Chapter contents	457
Determining the Level of Your !DB/EXPLAIN System	458
!DB/EXPLAIN Online Messages	459
Assistance from Candle Customer Support	461
Using the PROFOUT Command	463
Appendixes	465
Appendix A. Additional Problem Resolution Techniques	467
Using the SNAP Commands	468
Using the TRACE Commands	470
Interactively Tracing Statement Execution	471
Using the Automated Trace Facility	472
Diagnosing Parser Interpretation	474
Appendix B. Running !DB/EXPLAIN Concurrently in Batch and Online	477
Recommendations for Concurrent Access	480

Appendix C. Candle Customer Support	483
Customer Support Phone Numbers	485
Index	487

Preface

Efficient SQL is the foundation for optimal DB2 performance. !DB®/EXPLAIN for DB2 provides answers to SQL performance questions.

!DB/EXPLAIN provides developers with the means to more easily write efficient SQL code for high performance applications.

Using !DB/EXPLAIN, database administrators can perform the analysis needed to ensure that inefficient SQL does not degrade DB2 application or subsystem performance.

When used in conjunction with OMEGAMON II® for DB2, !DB/EXPLAIN provides a means for system programmers and DBAs to view access path data provided by !DB/EXPLAIN together with application trace and accounting data provided by OMEGAMON II for DB2 to quickly pinpoint poorly performing SQL and improve application performance.

This *!DB/EXPLAIN User's Guide* describes the features of !DB/EXPLAIN and guides you in performing common !DB/EXPLAIN tasks.

An associated publication, the *!DB/EXPLAIN Reference* illustrates !DB/EXPLAIN's major object list panels and describes the fields each contains. It also describes the commands and selects available for use on each panel, and the formats in which each panel can be displayed. The panels that support Housekeeping and Customization are also described.

Contents of this Book

Overview

This unit describes the contents of each major section of the manual.

Content of major sections

Use the table to understand the organization and content of this guide.

Section name	Contents
Preface	The Preface describes the customer support available from Candle® Corporation, the documentation available with !DB/EXPLAIN, and the conventions used in the !DB/EXPLAIN publications.
Getting Started	This section introduces you to !DB/EXPLAIN.
Introducing Fundamentals	This section tells you how to access and exit !DB/EXPLAIN, how to use basic !DB/EXPLAIN functions, how to use commands, selects, sorts, and filters, and how to access alternate formats of !DB/EXPLAIN's object list panels. The section also includes a short review of ISPF fundamentals for those that require it.
Using !DB/EXPLAIN System Facilities	Use this section to understand how to use !DB/EXPLAIN's macro facility and how to use !DB/EXPLAIN with OMEGAMON II for DB2 to improve system performance. This section also contains information about using !DB/EXPLAIN in a distributed environment.

Content of major sections (continued)

Section name	Contents
Performing !DB/EXPLAIN Tasks	This section provides scenarios of common !DB/EXPLAIN tasks. The scenarios can be used to learn more about !DB/EXPLAIN.
Customizing !DB/EXPLAIN	This section describes ways in which you can customize !DB/EXPLAIN to meet the needs of your installation.
The Extract Program	This section describes the extract program, how to use it, and how to determine storage requirements.
Using Batch Facilities	This section describes batch support for application development and database administration.
Problem Resolution	Use this section to resolve problems and report any incidents to Candle's support staff.
Appendixes	Use the appendixes to learn more about !DB/EXPLAIN problem resolution and running !DB/EXPLAIN concurrently in batch and online modes.

Adobe Portable Document Format

Introduction

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

1. Specify the print options for your system. From the Acrobat Reader Menu bar, select **File > Print Setup...** and make your selections. A setting of 300 dpi is highly recommended as is duplex printing if your printer supports it.
2. To start printing, select **File > Print** on the Acrobat Reader Menu bar.
3. On the Print popup, select one of the **Print Range** options for
 - a single page
 - a range of pages
 - all of the document
4. (Optional) To fit oversize pages to the paper size currently loaded on your printer, select the **Shrink to Fit** option.

Printing problems?

Your printer ultimately determines the print quality of your output. Sometimes printing problems can occur. If you experience printing problems, potential areas to check are:

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

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

Documentation Conventions

Overview

This guide uses the conventions described in this unit.

Convention Description

Refer to the table for a description of the documentation conventions used in this guide.

Convention	Description
Panels and figures	Panels and figures in this document are representations. Actual product panels may differ.
Revision bars	Revision bars (!) appear in the left margin to identify new or updated material.
Commands	DB2, TSO, ISPF, and !DB/EXPLAIN command names are in upper case.
Percent sign	When executing a CLIST, you can optionally precede the CLIST name with a percent sign (%) to achieve optimum performance. For example, to access !DB/EXPLAIN, you can invoke the CLIST KTE in the form KTE or %KTE .
Square brackets	Square brackets [] denote optional arguments. Arguments enclosed in square brackets are not required. In the following example, use of XLV is optional: [XLV]
Braces	Braces { } denote required arguments. In the following example, the <i>workload</i> keyword is required: MONITOR {workload} [time] [SUMMARY]

Convention Description (continued)

Convention	Description
<i>hilev</i>	<p>Lower-case italics in data set names denote variable qualifiers. Qualifiers are prefixes in a data set name that make the data set name unique. Some qualifiers used in this document are:</p> <p><i>hilev</i> A high-level qualifier; the first prefix or set of prefixes in the data set name.</p> <p><i>db2id</i> The DB2 subsystem identifier; used as a mid-level qualifier for some !DB@/Tools for DB2 data sets. The mid-level qualifier is the prefix or set of prefixes between the high-level qualifier and the last part of the data set name.</p> <p><i>extract_id</i> An extract ID; an identifier for the catalog data extracted from the DB2 catalog. It is used as a mid-level qualifier for some !DB/Tools data sets. The mid-level qualifier is the prefix or set of prefixes between the high-level qualifier and the last part of the data set name.</p> <p><i>product_id</i> A product ID; an identifier for the !DB/Tools product or products you are installing. It is used as a mid-level qualifier for some !DB/Tools data sets. The mid-level qualifier is the prefix or set of prefixes between the high-level qualifier and the last part of the data set name.</p>
'tddoc3.booklib.prod'	<p>Single quotes distinguish specific data set names from variable qualifiers. The data set name is shown in lower case and in the same font as the rest of the text.</p>
Function keys	<p>This guide does not refer to specific function keys, for example, F3. Because you may use the function key definitions provided by ISPF or choose to define your own function keys, this guide instructs you to enter a command on the command line or press the appropriate function key.</p> <p>Example: Type END on the command line and press Enter, or press the appropriate function key.</p>

Terminology—panel versus display

In this document, the term “panel” is used to denote a facsimile of an actual screen you might see when logged onto !DB/EXPLAIN. The term “display” is used to denote the information that is contained on a panel or the equivalent information written to an output device such as a printer or output file.

This distinction is especially important in the chapters of this manual that discuss the use of !DB/EXPLAIN in batch. In these chapters, you are asked to construct batch jobs to perform processing equivalent to entering commands, filters, and sorts on the command line of an object panel. Thus:

- A discussion of the “EXPLAIN display” refers to the EXPLAIN information presented either as a series of panel images or as information contained in a file that can be output to a printer.
- A discussion of a batch job containing commands usable from the “Statements display” implies that you can use only those commands you might use in an online session from the Statements panel.

Documentation Set

Introduction

Candle provides a complete set of documentation for !DB/EXPLAIN. Each manual in this documentation set contains a specific type of information to help you use the product.

Candle welcomes your comments and suggestions for changes or additions to the documentation set. A user comment form, located at the back of each manual, provides simple instructions for communicating with Candle's Information Development department. You can also send email to **UserDoc@candle.com**. Please include the product name, version, and book title in the subject line. To order additional manuals, contact Candle Customer Support.

The documentation listed in the following table is available for !DB/EXPLAIN.

Table 1. !DB/EXPLAIN Documentation		
Doc Number	Title	Description
TE54-5842	<i>!DB/EXPLAIN User's Guide</i>	Tells how to use !DB/EXPLAIN and its features
TE53-5843	<i>!DB/EXPLAIN Reference</i>	Provides a reference for major !DB/EXPLAIN functions and features
TI51-5840	<i>!DB/Tools Installation and Customization Guide</i>	Tells how to install !DB/EXPLAIN as well as the other !DB®/Tools for DB2 products. It also contains information about security and authorizations.

Online documentation for BookManager

In addition to the printed versions, !DB/EXPLAIN documentation is available in BookManager format. Both major !DB/EXPLAIN releases and maintenance releases between major releases contain updated BookManager documentation. See the *!DB/Tools Installation and Customization Guide* for information regarding these files.

For more information about using BookManager, see these IBM documents:

IF you want to ...	THEN see ...
put our book on a bookshelf,	<i>IBM BookManager BUILD/MVS Preparing Online Books SC38-2036</i>
display our book,	<i>IBM BookManager READ/MVS Displaying Online Books SC38-2034</i>

Version 500 Changes

Overview

This section describes what is different between !DB/EXPLAIN and previous versions of !DB/EXPLAIN.

Differences in this release

| Primarily, the Version 500 release of !DB/EXPLAIN provides full
| compatibility for DB2 Version 6. However, some DB2 enhancements are not
| exploited by !DB/EXPLAIN Version 500.

DB2 Version 6 contains enhancements for

- defining and manipulating data objects
- conducting e-business
- improving performance and availability of database applications
- managing the database environment
- increasing database and query capacity

This release of !DB/EXPLAIN also continues to support fully earlier releases of DB2 and is also Y2K compliant.

Getting Started

Chapter 1. Introducing !DB/EXPLAIN

Introduction

This chapter introduces you to !DB/EXPLAIN's major functions and the benefits !DB/EXPLAIN can provide in your installation. Read this chapter for a quick overview of the product. If you want to find more information about a specific topic, pointers in the text guide you to the details.

Chapter contents

Summary of !DB/EXPLAIN's Major Functions	26
Benefits of !DB/EXPLAIN's Extract Technology	27
Benefits for Database Administration	28
Benefits for Application Development	29
Support for Improved System Performance	31

Summary of !DB/EXPLAIN's Major Functions

Overview

!DB/EXPLAIN assists DB2 database administrators, application programmers, and system programmers by increasing their productivity and efficiency. This unit introduces !DB/EXPLAIN facilities. Subsequent units in this chapter describe the benefits provided to !DB/EXPLAIN's users.

The extract

!DB/EXPLAIN's flexible and efficient batch extract derives its data from the DB2 catalog. Once you have performed an extract, you can access the extracted data an infinite number of times without impact to your active DB2 system. The availability of user-specified extract IDs permits you to perform multiple extracts against the DB2 catalog, each extract “configured” to meet your needs by means of !DB/EXPLAIN's extensive masking capabilities.

Online facilities

Online panels help you to manage and monitor DB2 objects, analyze SQL code, perform “what if” analysis, and view and compare histories.

Batch facilities

Because most functions of !DB/EXPLAIN are available in both online and batch mode, you can perform any resource-intensive tasks in a way that causes the least impact to your online users and active system.

!DB/EXPLAIN and OMEGAMON II for DB2

You can integrate !DB/EXPLAIN functions with those of Candle's OMEGAMON II for DB2 realtime performance monitor. Combining OMEGAMON II's runtime performance data with predictive analysis data from !DB/EXPLAIN helps you quickly pinpoint and fix poorly performing SQL.

Benefits of !DB/EXPLAIN's Extract Technology

Overview

!DB/EXPLAIN gets its information from batch extracts which can be run during periods of low use to conserve DB2 and system resources. This unit describes the benefits you gain through your use of the extract.

Benefits

!DB/EXPLAIN's extract technology is flexible. You can extract all, or a portion of, the data in your DB2 catalog so that your catalog information is as up-to-date as your needs require. You can limit the extracted data according to criteria you specify. You can also extract data from PLAN_TABLES, and, if you wish, combine data from multiple PLAN_TABLES from the same DB2 subsystem.

Because !DB/EXPLAIN sorts extract data external to DB2, you obtain improved performance and efficiency of the sort.

Because !DB/EXPLAIN acts on an extract rather than directly on the DB2 catalog:

- You can access extract data even when DB2 is not active.
- You avoid contention on the DB2 catalog.
- You can retain histories of plan, package, and DBRM catalog data and their EXPLAINS.

!DB/EXPLAIN's extract provides significant benefit to database administrators and application programmers as well. These benefits are discussed in the units that follow. For detailed information about !DB/EXPLAIN's extract, see "The !DB/EXPLAIN Extract Program and the Extract ID" on page 333.

Benefits for Database Administration

Overview

This unit describes the benefits !DB/EXPLAIN provides to those performing the task of database administration.

Reviewing and improving SQL

!DB/EXPLAIN enables a database administrator to review application code and monitor application-specific SQL performance. You can EXPLAIN your SQL code to understand the DB2 access paths taken and their relative cost. You can specify exception categories for which !DB/EXPLAIN generates reports. Such exceptions might be SQL statements that cause DB2 table space scans, sorts, or locking.

!DB/EXPLAIN permits you to store EXPLAINs with the SQL they reference, the associated costs, and the statistics. You can use this historical information to perform impact analysis when changes to an application are proposed. You can also perform historical analyses by comparing EXPLAIN histories.

You can temporarily or permanently modify the results of RUNSTATS to perform “what-if” analysis, observing any effect of changes to access path selection.

Managing DB2 objects

!DB/EXPLAIN provides extensive cross-reference reporting capabilities, permitting you to view such things as statement use by package or plan, DBRMs by plan, packages by plan, and DBRMs or plans accessing tables. You can BIND, REBIND, or FREE DB2 objects using !DB/EXPLAIN facilities. You can review DB2 objects to understand their attributes such as BIND date and owner. You can also use !DB/EXPLAIN to convert plan DBRMs to packages.

Using !DB/EXPLAIN to facilitate migration

When migrating from a previous version of DB2 to the current version, you can use current EXPLAINs together with EXPLAIN history to view and compare access paths to determine where performance tuning is required during migration.

!DB/EXPLAIN makes apparent the influence that the SQL code has on the DB2 optimizer's decisions. Those SQL statements that generate the most efficient code are easily identified.

Benefits for Application Development

Overview

Because inefficient SQL code directly affects DB2 performance, SQL performance tuning is very important. This unit describes the benefits !DB/EXPLAIN offers to the application developer.

Tuning SQL code

!DB/EXPLAIN is an invaluable tool for the SQL programmer during application development. You can use !DB/EXPLAIN to tune your SQL before finalizing your design and implementing your application. !DB/EXPLAIN helps you develop efficient SQL code by:

- Permitting you to EXPLAIN your SQL code within an interactive ISPF session or ISPF edit session to obtain access path information

Detailed information on how application developers can access !DB/EXPLAIN interactively to EXPLAIN their SQL code prior to its being bound can be found in “Access from an Edit Macro” on page 77 and “Access Using the EXEC KTEON” on page 90.

- Identifying potential trouble spots in the selected access path
- Permitting you to execute your SQL code within an interactive ISPF session or ISPF edit session to view and validate the results set
- Providing a batch facility within which you can execute your SQL code and obtain a report

Using batch !DB/EXPLAIN helps to conserve system resources while providing extensive diagnostic information. You can find more information about using batch !DB/EXPLAIN for application development in “Batch Processing for Application Development” on page 419.

- Permitting direct access from IBM's Query Management Facility (QMF) to EXPLAIN SQL queries, prompted queries, and query-by-example (QBE) queries

Improving SQL coding techniques

In addition to assisting you with tuning your SQL code, using !DB/EXPLAIN may help you improve your SQL coding techniques. !DB/EXPLAIN can serve as a teaching tool by:

- EXPLAINing the results of the execution in plain English
- Suggesting more efficient alternative techniques for coding SQL

Comparing !DB/EXPLAIN's predicted access path information with the access path recommendations !DB/EXPLAIN generates can “teach” you improved SQL coding techniques.

Support for Improved System Performance

Overview

This unit describes the benefits of using !DB/EXPLAIN and Candle's OMEGAMON II for DB2 together to improve your DB2 system performance.

Benefits of OMEGAMON II for DB2 integration

The close coupling of !DB/EXPLAIN and Candle's OMEGAMON II provides you with a means to identify and change problem SQL in a timely manner. While in a !DB/EXPLAIN session, you can display OMEGAMON II for DB2's application trace and plan accounting reports. Alternatively, you can access !DB/EXPLAIN from an OMEGAMON II session.

The tight integration of !DB/EXPLAIN and OMEGAMON II for DB2 permits you to quickly pinpoint poorly performing SQL and make appropriate changes. A “copy-and-paste” feature of OMEGAMON II captures the SQL code you identified and allows you to move it into !DB/EXPLAIN for tuning.

The dynamic prefetch and I/O parallelism facilities of DB2 may make predictive analysis difficult. In these cases, you may want to use OMEGAMON II's application trace output within a !DB/EXPLAIN session to see the EXPLAIN and the trace together. This permits you to understand how the SQL call actually performs in the production environment. You can find more information on the integration of !DB/EXPLAIN and OMEGAMON II for DB2 in “Using !DB/EXPLAIN with OMEGAMON II for DB2” on page 191.

Chapter 2. Introducing !DB/EXPLAIN Information

Introduction

Information about !DB/EXPLAIN is available to you in several forms.

- This guide
- Online help
- Associated publications

This chapter describes the information available to assist you in using !DB/EXPLAIN and describes how the information works together to provide the best possible support for performing !DB/EXPLAIN tasks.

Chapter Contents

Structure of this Guide	34
Using !DB/EXPLAIN Online Help	36
Elements of a !DB/EXPLAIN Help Panel	37
Using Associated Publications	40

Structure of this Guide

Overview

This unit shows you how you can use this guide to increase your productivity.

How the structure of this guide improves your productivity

As you become accustomed to the structure of this guide, you can improve your productivity in the following ways:

- Quickly access pertinent information you need
 - Avoid viewing information that is extraneous to the task you want to perform
-

Sections

The major sections of this guide begin with a title page and are separated by tabs. Tabs assist you to locate these broad topics easily. The guide contains a number of major sections. You are currently using the section called “Getting Started.”

Chapters

Within each section, topics are organized into chapters. A chapter contains small, related task-based units of information. The title page of each chapter introduces the information found in the chapter and provides a table of contents so you can locate specific units easily.

Units

This guide is organized into small, task-based units containing one or more pages of information. Each unit has its own title that appears as a running heading at the top of all of the pages comprising the unit. For example, the title of this unit, “Structure of this Guide,” can be seen above.

Each unit starts with an overview. Reading the overview quickly tells you whether the unit contains the information you need. Thus, the amount of information you need to read is reduced.

Locating information you need

The chart shows the devices provided in this guide to help you locate the type of information you want.

Task you want to perform	Device to use
Understand the information provided by this guide	Skim the table of contents. (In this guide, the table of contents includes titles of units within each chapter so that you can determine the tasks covered in a chapter.)
Locate specific information about a task	Either: <ul style="list-style-type: none"> ● Skim the table of contents ● Consult the alphabetic index at the back of the book
Find panel images for !DB/EXPLAIN's object list and housekeeping panels	Consult the chapter called "Panel Reference"
Quickly locate reference information about commands, selects, filters, or display fields	Either: <ul style="list-style-type: none"> ● Skim the titles of the appendixes (located at the end of the book) in the table of contents ● Consult the alphabetic index at the back of the book

Using !DB/EXPLAIN Online Help

Overview

This unit contains brief instructions for using the online Help available with !DB/EXPLAIN.

Categories of online Help

Online Help falls into three categories:

1. Help for !DB/EXPLAIN panels
 2. Help for ISPF facilities
 3. Help for messages
-

Background about !DB/EXPLAIN online Help

In !DB/EXPLAIN, you can use online Help to access specific information about each panel. Help is available to:

- Describe the contents of a specific field on a specific panel, if the field is a display field
 - Describe the values that may be input to a given field, if the field is one that can accept input
 - Provide the meaning for the selects available for a given panel
 - Describe the functions of the global and local commands available for a given panel
 - Describe the functions of the sorts and filters available for a given panel
-

Accessing online Help from a !DB/EXPLAIN menu or panel

Follow this step to access online Help.

Step	Action
1	<p>On a !DB/EXPLAIN menu or panel, type HELP on the Command line and press Enter, or press the appropriate function key.</p> <p>Result: !DB/EXPLAIN displays the Help panel associated with the menu or panel you are using.</p>

Elements of a !DB/EXPLAIN Help Panel

Overview

This unit illustrates the elements of a typical !DB/EXPLAIN Help panel.

Types of Help

There are two major types of Help panels in !DB/EXPLAIN. The first type provides general Help for a given panel and provides quick access pointers to the balance of the Help. The second type can be accessed by pointers from the general Help and describes the functions and fields on an object list panel. For data entry panels, the second type lists the possible values that can be entered.

General Help

This is a typical general or overview help panel.

```
----- DB/EXPLAIN DB2=D31A -----
CMD ==>
1 The Plans panel displays a list of plans defined in the DB2 Catalog.
2 Formats
The Plans panel is available in five formats. You can set your own
default format and switch back and forth between formats, making
sure you are seeing the plan information that's important to you.
3 Actions Available
The Plans panel allows you to sort, filter, EXPLAIN, and BIND
plans on the display list. You may compare a plan with a dummy
plan or an existing plan, display a listing of packages,
package lists, collections, system connections, costs, and
tables for a selected plan.
Press END to exit help or choose one of the following:
  1. Available Commands
  2. Available Fields
  3. Available Filters
  4. Available Formats
  5. Available Selections
  6. Available Sorts
  7. Global Commands
```

- 1** General description of the purpose or content of the object list panel.
- 2** Description of available formats for the object list panel (if applicable)
- 3** Description of the actions you can take on the object list panel.
- 4** Quick access pointers to the balance of the Help.

Field Help

This is a typical field Help panel.

1	2	3
Field Name DEGREE	Description The DEGREE option used when the plan was last bound. (Indicates whether or not DB2 attempts to execute queries with parallel I/O processing.) ANY Allow parallel I/O 1 Prohibit parallel I/O blank Migrated plan Prohibit parallel I/O	Source SYSPLAN.DEGREE
SERVER	Location name specified with the CURRENTSERVER option when the plan was last bound. Blank implies that the first server is the local DB2 subsystem.	SYSPLAN.CURRENT SERVER

Press ENTER for more or END to exit help...

Field help panels may be of two types. In some instances, a field is a display field. In that case, the Help panel describes the displayed content. At other times, a field may be a data entry field. In that case, the Help panel provides the possible values that can be entered into the field.

- 1** Name of the field as shown on the object list panel.
- 2** Description of the field's contents
- 3** On this panel, which displays information taken from the DB2 catalog, this is the name of the DB2 catalog table that is the source of the values in the field. On a data entry panel this field would contain the values that are allowable for entry into the field.

Help for ISPF facilities

Use this step to access Help for ISPF facilities.

Step	Action
1	On a !DB/EXPLAIN Help panel, type HELP on the Command line and press Enter, or press the appropriate function key. Result: !DB/EXPLAIN displays a general ISPF Help panel that discusses ISPF facilities.

Help for messages

If a processing error should occur when you are using !DB/EXPLAIN, an error message is displayed in the upper right hand corner of the panel. To obtain expanded information about the error message, review this chart.

Step	Action
1	On a !DB/EXPLAIN menu or panel that is displaying the error message, type HELP on the Command line and press Enter, or press the appropriate function key. Result: !DB/EXPLAIN displays expanded information about the error condition.
2	Some messages have another level of Help. When the long message is displayed, you can press the appropriate function key again. If additional help is available, it will be displayed.

Using Associated Publications

Overview

This unit describes the associated publications available with !DB/EXPLAIN.

Publications list

This chart lists the associated publications available with !DB/EXPLAIN.

Associated publication	Purpose or use
!DB/Tools Installation Guide	This document provides detailed instructions on installing !DB/EXPLAIN and other Candle data base tools. It also provides information about running the !DB/EXPLAIN extract.
!DB/EXPLAIN Version Release Guide	This document highlights major features of a new version or release of !DB/EXPLAIN.
Technical Bulletins (optional)	A technical bulletin may be issued to support a small programming enhancement or other small technical changes to !DB/EXPLAIN when documentation is needed but the volume of material does not warrant reissuing the User's Guide.

Chapter 3. !DB/EXPLAIN Basics

Introduction

This chapter introduces you to the basic activities you can perform using !DB/EXPLAIN. These activities include performing an EXPLAIN, binding or rebinding objects, comparing objects, and displaying the results of actions against objects. The activities also include the ability to perform temporary updates to a local or remote DB2 catalog, and the ability to manipulate catalog statistics to obtain the optimum access path for your SQL statements. As noted earlier in this Guide, most of the activities you can perform are available for both online and batch execution.

Information in this chapter, as well as subsequent chapters of this manual, assumes you are familiar with DB2 and its constructs and with the SQL language.

A description of each of the !DB/EXPLAIN panels cited in this unit can be found, listed alphabetically by panel name, in the *!DB/EXPLAIN Reference*. The description provides:

- Information on accessing the panel
- A description of the fields that comprise the panel
- A description of the commands and selects you can use from the panel

Chapter contents

What is an EXPLAIN?	42
Setting Up Your EXPLAIN Output	44
Elements of Your EXPLAIN Output	46
The BIND Section	47
The Statements Section	49
Running an EXPLAIN	54
EXPLAINing SQL Statements	56
Displaying !DB/EXPLAIN Information	57
Comparing Information About Objects	59
Reviewing Exception Conditions	61
Changing Values and Playing What-if	62
Using !DB/EXPLAIN Facilities to Generate JCL for Common Tasks	63
Binding or Rebinding Objects	64
Other Basic Functions You Can Perform	66

What is an EXPLAIN?

Overview

This unit describes the basic purpose for running an EXPLAIN and the output of an EXPLAIN. It also identifies requirements for running an EXPLAIN.

Purpose of an EXPLAIN

Running an EXPLAIN provides you with information about the access path the DB2 optimizer takes when executing an “explainable” SQL statement. A statement is explainable if it is a SELECT or INSERT statement or if it is an UPDATE or DELETE statement without a “where current of” clause. Information resulting from running an EXPLAIN is placed into a plan table that you must supply.

Objects that can be EXPLAINed

One primary function of !DB/EXPLAIN is to EXPLAIN the SQL statements associated with DBRMs, packages, and plans. You can EXPLAIN individual SQL statements you select from a display of statements comprising a DBRM, package, or plan. You can also EXPLAIN BIND commands (all DBRMs that would be included in the plans, packages, or both that would be bound by a BIND command).

Output of an EXPLAIN

!DB/EXPLAIN interprets the information stored in the !DB/EXPLAIN plan table and presents it to you in various formats, including a format you can tailor to meet the specific needs of your site. !DB/EXPLAIN provides a significant advantage over the use of DB2's explain facility in these ways:

- !DB/EXPLAIN presents access path information to you in English rather than as numeric codes.
- !DB/EXPLAIN presents comprehensive recommendations for changes to your SQL to improve performance. You can tailor the recommendations that !DB/EXPLAIN displays as well as the amount of information displayed when a recommendation is triggered.
- !DB/EXPLAIN provides extensive tabular information about the characteristics of the object being EXPLAINed.
- You can alter catalog statistics and observe the effect of these alterations by means of running and comparing EXPLAINs. You can then make the alterations permanent or you can discard them.

Using EXPLAIN data in a real-time environment

Because of factors inherent to DB2, such as sequential prefetch or I/O parallelism, the access path that !DB/EXPLAIN predicts may vary from real-time execution. To assist in the SQL tuning process, !DB/EXPLAIN can be used in conjunction with OMEGAMON II for DB2 to compare actual run time access with !DB/EXPLAIN's predicted access. The use of these products together is described in the chapter "Using !DB/EXPLAIN with OMEGAMON II for DB2" on page 191.

Setting Up Your EXPLAIN Output

Overview

This unit describes how you can define the content of the output that !DB/EXPLAIN will produce.

Background about setting up EXPLAIN output

The output you receive for your system will differ depending on the characteristics of your DB2 subsystem, the object you are EXPLAINing, and whether or not you have customized your EXPLAIN output using the EXPLAIN Display Tuning Parameters panel.

About the EXPLAIN Display Tuning Parameters panel

This is an example EXPLAIN Display Tuning Parameters panel. For information about using it to customize your EXPLAIN output, see “Controlling the Format of the EXPLAIN Display” on page 303.

```

----- DB/EXPLAIN DB2=D31A -----
Cmd ==>

  E X P L A I N   D I S P L A Y   T U N I N G   P A R A M E T E R S

PERMANENT? ==>  ( Y Yes  N No )

                3          5          4
                Generate? Format0  Order
                Y          Y          1
EBIND information
ERECM (Bind Specific)
ECOST information      Y          Y          3  ] 1
                Y          Y          5
EOBJ information
EKEYS information     Y          Y          3
EPATH information     Y          Y          2  ] 2
EPTBL information     Y          Y          4
EPTB2 information     Y          Y          1
ERECM information     Y          Y          1
ESTMT information     Y          Y          1

ENTER to process  END to cancel

```

- 1** Parameters that define the BIND section of the EXPLAIN display. Recommendation information in the BIND section does not require a specification.
- 2** Parameters that define the Statements section of the EXPLAIN display
- 3** Column that defines whether a specific type of EXPLAIN output is generated. To cause information of this type to be generated, type Y in this column. Information cannot be displayed on an EXPLAIN unless you first generate it.
- 4** Specification that determines the order in which information appears on your EXPLAIN display
- 5** Column that identifies the information to be displayed on your customized EXPLAIN display. To cause information of this type to be displayed, type Y in this column. The information is displayed in the order you define in the last column of this panel.

Elements of Your EXPLAIN Output

Overview

This unit describes the various components of the output of an EXPLAIN.

Elements of an EXPLAIN

The EXPLAIN display comprises two sections:

- The *BIND section* comprises BIND information, BIND-specific recommendations, and summary cost information.
 - The *statements section* comprises statement information, plan table information, information about statement cost, access path strategy, and recommendations, and detailed information about the objects being EXPLAINed.
-

About this example

The example you will review on the pages that follow describes the output of an EXPLAIN for a plan. It is produced using format 0, a format you define using the EXPLAIN Display Tuning Parameters Housekeeping panel. Using this panel, you can define what information you want to display and the order in which you want the information to be displayed.

The BIND Section

Overview

This unit describes the BIND section of an EXPLAIN display.

Reviewing the BIND section

The BIND section of an EXPLAIN contains BIND-type information for the object being EXPLAINed, recommendations for the object, and summary cost information. These panels illustrate the components of the BIND section of an EXPLAIN for plan DSNHYCRD. Detailed information about individual fields on any of the illustrations can be found in “EXPLAIN” in the *!DB/EXPLAIN Reference*.

```

----- DB/EXPLAIN DB2=D42B ----- LINE 1 OF 222
Cmd ==> FORMAT1                               SCROLL ==> PAGE
                E X P L A I N
  Cnds: DO (Menu) GLOBAL (Menu)
-----
PLAN=DSN8BD42  DBRM=DSN8BD3  CONTOKEN=15B3C90A0B7E75A0
PRECOMPILE=1999/10/14 16:41:17.260000
EXPLAINED=1999/12/09 11:46:04  DEGREE=1
+----- DETAILED PLAN INFORMATION -----+
| PLAN NAME = DSN8BD42  CREATOR = DCYBU2  PLENTRIES =    0 |
| PLAN FREED? = N      QUALIFIER= DCYBU2  SYSENTRIES=    0 |
| AVERAGE SIZE=   3194  BOUND BY = DCYBU2  DBRMS      =    1 |
| PLAN SIZE      =   2728  BIND DATE= 1999/10/14  PACKAGES =    0 |
| CACHE SIZE    =   1024  BIND TIME= 16:42:01  GROUP MEMBER= |
| DEGREE        =    1    SERVER      = |
|
| VALIDATE = R  ISOLATION = S  VALID = Y  OPERATIVE = Y  ACQUIRE= U |
| EXPLAIN  = N  DEFERPREP = N  SQLRULES= D  DISCONNECT= E  RELEASE= C |
| EXPREDICATE= C  DYNAMICRULES=  KEEPDPY= N  REOPT    = N |
+-----+
+----- RECOMMENDATIONS FOR DBRM DSN8BE3 -----+
| RECOMMENDATION B00004 |
| The plan or package this statement belongs to was bound with the |
| option VALIDATE(RUN). Generally the option VALIDATE(BIND) is |
| preferred. |
|
| When the option VALIDATE(RUN) is specified, DB2 does validity |
| checking at BIND time and then rechecks any failures at run time. |
| The validity checking at run time can cause catalog contention |
| and degraded performance. If possible, ensure that all objects |
| exist and all privileges are granted prior to binding a plan or |
| package and then choose the VALIDATE(BIND) option. |
+-----+
+----- STATEMENT NUMBER 286 TEXT-----+
| DECLARE TELE1 CURSOR FOR |
| SELECT * |
| FROM VPHONE |
+-----+
+-----STATEMENT NUMBER 286 ERROR-----+
| DSNT4081 SQLCODE = -204, ERROR: DCYBU2.VPHONE IS AN UNDEFINED NAME |
| DSNT4181 SQLSTATE = 42704 SQLSTATE RETURN CODE |
+-----+

```

Understanding the cost information

In looking at the cost information for the various EXPLAINable statements that the plan contains, note that the SQLCODE field for each statement is 0. A message indicates that the cost *may* be in error. Similarly, when the statement cost is displayed for each individual statement, that cost is suffixed with an asterisk (*). This condition can occur when a column is compared to a host variable or literal having a different type or length.

The Statements Section

Overview

This unit describes the statements section of an EXPLAIN display.

Reviewing the statements section

The statements section of an EXPLAIN contains information specific to each EXPLAINable statement. The balance of this unit describes the statement information for statement 1047 of plan DSNHYCARD. Detailed information about individual fields on any of the illustrations can be found in “EXPLAIN” in the *!DB/EXPLAIN Reference*.

Statement text

This statement text information for statement 1047 follows. Note that this statement contains host variables denoted by the colon (:).

```

+----- STATEMENT NUMBER 1047 TEXT -----+
| DECLARE CURSOR2 CURSOR FOR                |
| SELECT NAME , COLTYPE , LENGTH , SCALE , NULLS , COLNO |
| FROM SYSIBM.SYSCOLUMNS                    |
| WHERE TBNAME = :INNAME AND TBCREATOR = :INAUTH |
| ORDER BY COLNO                             |
+-----+

```

Access path summary information

Access path summary information is printed for each step in determining the access path. The summary information includes:

- The name of the tables) being accessed and any indexes being used
- The statement cost
- Whether the EXPLAIN was created by an EXPLAIN request, a plan table extract, a BIND compare, or as the result of a what-if process
- Whether or not a join was required
- The method DB2 is assumed to use to access the table
- Whether an index was used
- Various fields related to query I/O parallelism
- If a new or composite table was created, whether a sort on that table was required
- The lock mode of the table space
- Whether prefetch is anticipated to be used

Access path summary information (continued)

```

CMDS: DO (MENU) GLOBAL (MENU)
-----
PACKAGE=DSNHYCRD      COLLID=DSNHYCRD      CONTOKEN=1554036E608555E10
PRECOMPILE=1999/02/13 07:38:37.843832
EXPLAINED=1999/12/09 06:53:19  DEGREE=1
+----- ACCESS PATH SUMMARY -----+
| STMT QB  PLN MX EMJP AC I  MT MJ P ACCESS JOIN  SRNSRC SRN SRC  PFW |
| NO. NO. NO. OP METR CS X  CL CL M  DG ID  G ID  ID  ID UJOGUJOG LCK FNO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1051  1  1  0 E0  I  N  2  0  0  0  0  0  0  0  0  0  NNNNNNNN  IS  L  |
| 1051  1  2  0 E3  N  0  0  0  0  0  0  0  0  0  0  NNNNNNNN  |
| 1056  1  1  0 E0  I  N  2  0  0  0  0  0  0  0  0  0  0  NNNNNNNN  IS  L  |
| 1056  1  2  0 E3  N  0  0  0  0  0  0  0  0  0  0  NNNNNYNN  |
| 204  1  1  0 E0  I  N  2  0  0  0  0  0  0  0  0  0  0  NNNNNNNN  IS  L  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
    
```

The access path strategy

The information summarized by the access path summary display is expanded on in the access path strategy portion of the EXPLAIN.

```

+----- ACCESS PATH STRATEGY FOR STATEMENT NUMBER 1047 -----+
Statement Cost: 15.1
The cost (timerons) is a rough estimate of resources required to
execute the SQL statement.
The cost may be in error due to mismatching column and host variable
data types or lengths.

Query Block No. 1

Step 1

Access Info:

This is either a composite table or it is the first table accessed.
This step accesses SYSIBM.SYSCOLUMNS.
Uses index SYSIBM.DSNDCX01.
Matching on 2 index columns.
Table pages are accessed.
Column functions are to be decided at run time.
Query type: SELECT

Lock Info:

Intent Share. Lock holder has read-only access.
Concurrent reads and updates are allowed.

Step 2

Access Info:

This step sorts for ORDER BY, GROUP BY, SELECT DISTINCT or UNION.

Sort Info:

A sort is performed on composite tables to:
Order resulting rows
+-----+
    
```

The recommendations

!DB/EXPLAIN provides a series of recommendations panels as part of its Housekeeping facility. On these recommendations panels, you can specify the following for each recommendation that !DB/EXPLAIN displays:

- One of four levels of recommendation text: none, terse, verbose, or very verbose. This example illustrates both terse and verbose recommendations.
- Whether a given recommendation is to be treated as an exception condition when it is triggered
- For some recommendations, a threshold value that causes the associated recommendation to be displayed only when the threshold is met or exceeded
- For a batch EXPLAIN, a return code to be set when a given recommendation is triggered

See “Recommendations Menu” through “Recommendations Panel 9” in the *!DB/EXPLAIN Reference* for information about customizing recommendation information.

```
+----- RECOMMENDATIONS FOR STATEMENT NUMBER 1047 -----+
RECOMMENDATION S00006
  The host variable or literal being compared to
  column TBNAME of table SYSIBM.SYSCOLUMNS
  is longer than the column. An index may not be used, or the
  number of matching columns may be decreased.
  Consider making the host variable or literal of the equal
  or less length than the column.

  The access path reported for this statement may be incorrect
  due to the length mismatch.

RECOMMENDATION S00006
  The host variable or literal being compared to
  column TBCREATOR of table SYSIBM.SYSCOLUMNS
  is longer than the column. An index may not be used, or the
  number of matching columns may be decreased.
  Consider making the host variable or literal of the equal
  or less length than the column.

  The access path reported for this statement may be incorrect
  due to the length mismatch.

RECOMMENDATION X00010
  Sort for Group By
+-----
```

Object information

This portion of the EXPLAIN displays information about the table space, tables and indexes accessed during the execution of the statement.

```

+----- DETAILED OBJECT INFORMATION FOR STATEMENT NUMBER 1047 -----+
TS= DSNDB06.SYSDBASE      PARTITIONS= 0  LOCKRULE = A
NACTIVE= 540              OBID= 1      CLOSERULE= N  CREATOR = SYSIBM
NTABLES= 14              DBID= 6      IMPLICIT = N  CREATEBY= SYSIBM
SPACE = 0                PSID= 9      STATUS  = A  BPOOL = BP0
PGSIZE = 4              SEGSIZE= 0  ERASERULE= N  DSETPASS=
LOCKMAX= 0              STATSTS= 1999/04/12 14:56:02.636093

TBL= SYSIBM.SYSCOLUMNS      LOCATION=
RECLEN= 408              TBCREATR=          TYPE = T
CARD = 2698              TBNAME =          STATUS= X
NPAGES= 165             CREATEBY= SYSIBM   CKFLAG=
PCTPAGES = 30           VALPROC =          AUDIT =
KEYOBID = 115           EDPROC =          CHILD = 4
OBID = 20               CKRID =           PART = 1
COLCOUNT = 20          RBA1 = 000000000000  DATA =
KEYCOLUMNS= 3          RBA2 = 000000002150
PCTROWCOMP= 0           CREATETS= 1997/04/01 00:00:00.000000
CHECKS = 0              ALTERTS = 1998/06/24 16:08:32.743284
CLUSTERTYPE =           STATSTS = 1999/04/12 14:56:04.221070

----- USED INDEXES -----
IX= SYSIBM.DSNDCX01      CREATEBY = SYSIBM
FULLKEY = 2698          NLEVELS= 2        CLUSTERATIO= 94
FIRSTKEY= 28            NLEAF= 3          CLUSTERED = N
SPACE = 0               COLCOUNT = 3     CLUSTERING= N
PGSIZE= 4096           ERASERULE= N      INDEXTYPE =
SUBPAG= 1               CLOSERULE= N      DSETPASS =
OBID = 115             UNIQUE = P         INDEXSPACE= DSNDCX01
ISOBID= 116            STATSTS = 1999/04/12 14:56:05.239802
BPOOL = BP0

      KY          O N
USE SQ KEY COLUMN NAME  R L COLTYPE  LENGTH SCALE COLCARD
-----
**> 1 TBCREATOR        A N CHAR      8          28
      HIGH2KEY= TSL251      LOW2KEY= CLOVIS

**> 2 TBNAME           A N VARCHAR   18          118
      HIGH2KEY= VSTAFAC1    LOW2KEY= ALTERCON

      3 NAME            A N VARCHAR   18          632
      HIGH2KEY= XCPTRID     LOW2KEY= ACCESS_D

+-----+
***** BOTTOM OF LIST *****

```

Displaying index keys information as part of an EXPLAIN

This object information example contains index keys information. You can view index keys information on the EXPLAIN format 0 and format 3 displays.

Prerequisites to displaying index keys information

Before you can display index keys information, you must create the required data. The prerequisites for creating the data follow:

- Specify a value of 3 or greater in the Amount of Statistics to Gather field on the EXPLAIN Defaults Housekeeping panel.
- Specify a value of Y in the Generate? column for the EKEYS operand on the EXPLAIN Display Tuning Parameters Housekeeping panel.

Displaying index keys information automatically

Provided you have correctly specified the prerequisite Housekeeping parameters, !DB/EXPLAIN automatically displays index keys information on format 3 of the EXPLAIN display.

In order for !DB/EXPLAIN to automatically display index keys information on format 0 of the EXPLAIN display, you must specify a value of Y in the Format0 Display? column for the EKEYS operand on the EXPLAIN Display Tuning Parameters Housekeeping panel.

Displaying index keys information manually

You can manually display index keys information on the EXPLAIN format 0 display. Follow these steps:

Step	Action
1	Create the Housekeeping definitions discussed under Prerequisites.
2	Access the EXPLAIN format 0 display.
3	Type the command EKEYS on the command line.
4	Press Enter. Result: !DB/EXPLAIN redisplay format 0 and includes index keys information.

Running an EXPLAIN

Overview

This unit describes the additional information you need to run an EXPLAIN.

Authorization requirements for running an EXPLAIN

In order to run an EXPLAIN, you must have appropriate authorization. For example, if you are EXPLAINing an UPDATE statement, you must have *update authority* for those tables being updated. Information about authorization is found in the *!DB/Tools Installation and Customization Guide*.

How to run a new EXPLAIN

You can run a new EXPLAIN from various !DB/EXPLAIN panels using commands and selects. You can find definitions of all of the !DB/EXPLAIN commands and selects in appendixes to the *!DB/EXPLAIN Reference*.

- Issue one of these commands from the DBRMS, Packages, or Plans panel. Each of these commands generates an EXPLAIN, but the objects that are EXPLAINed may differ depending on the format of the command. Commands that begin with the letter G request that !DB/EXPLAIN gather catalog statistics. Commands that begin with the letter E request that !DB/EXPLAIN omit gathering statistics. Commands that end with the letter R apply to a remote DB2 subsystem.
 - EXPL, GEXPL (can also be issued from the Statements panel)
 - EXPLA, GEXPLA (can also be issued from the Statements panel)
 - EXPLAR, GEXPLAR
 - EXPLO, GEXPLO
 - EXPLOR, GEXPLOR
 - EXPLR, GEXPLR
- Issue one of these selects from the DBRMs, Packages, Plans, or Statements panel:
 - !DB/EXPLAIN performs a remote EXPLAIN and gathers statistics:
 - @
 - E
 - !DB/EXPLAIN performs a remote EXPLAIN and omits statistics-gathering.
 - #
 - U
- Issue the 8 or 9 select from the DBRMs, Packages, or Plans panel. Using the 8 select causes !DB/EXPLAIN to gather catalog statistics. Using the 9 select omits statistics-gathering.

How to review the latest existing EXPLAIN

You may want to review the latest existing EXPLAIN for an object and, perhaps, compare it with a new EXPLAIN. Doing so is particularly useful when you have made changes to an application or to your DB2 subsystem. To display the latest EXPLAIN for an object, issue the LEXPL command or the L select from the DBRMs, Packages, Plans, Statements, or Exceptions panel.

Information about comparing information about objects can be found later in this chapter in “Comparing Information About Objects” on page 59.

How to change catalog statistics and review the associated EXPLAIN

The What-if panel provides statistical information about the tables and indexes used by an SQL statement. You can use this panel to temporarily or permanently modify the DB2 catalog and, by running an EXPLAIN using the modified statistics, observe the effect of changes on DB2's access path selection.

Use these commands from the What-if panel to run an EXPLAIN:

- WIF, WIFL, WIFR
- WIFU, WIFUL, WIFUR

Using WIF or WIFL permits you to update the local DB2 catalog with new statistics and run an EXPLAIN. Using these commands gives you the option to make the changes permanent. WIFR performs the same function at a remote DB2 subsystem.

Using WIFU or WIFUL permits you to temporarily update the local catalog and EXPLAIN using the new statistics. WIFUR performs the same function at a remote DB2 subsystem.

Finding further information

You can locate additional information about running an EXPLAIN in these chapters or units:

- Example *scenarios* that address running an EXPLAIN either online or in batch are found in “Example Tasks” on page 239.
- The various *formats* of the EXPLAIN display are found in “EXPLAIN” in the *!DB/EXPLAIN Reference*.
- Explanations of each of the *fields* that comprise the EXPLAIN display are described in “EXPLAIN” in the *!DB/EXPLAIN Reference*.
- Additional details about the formats, fields, commands, and filters available when running an EXPLAIN can be found in the *online Help*.

EXPLAINing SQL Statements

Overview

This unit provides an overview of EXPLAINing SQL statements.

Background about EXPLAINing statements

You can use !DB/EXPLAIN to EXPLAIN individual SQL statements. The EXPLAIN can be initiated by selecting the statements to be EXPLAINed in one of two ways:

- Selecting the object whose statements you want to view from the DBRMs, Packages, Plans, or Tables panels.
- Identifying the SQL code to be EXPLAINed when that code is contained in an application program. In this case, you would access !DB/EXPLAIN using an appropriate EXEC, macro, or CLIST. (See “Accessing and Exiting !DB/EXPLAIN” on page 69 for information.)

In either case, !DB/EXPLAIN displays the Statements panel. The Statements panel lists the SQL statements for the object or the code you identified.

Using the Statements panel

From the Statements panel, you can use commands and selects to:

- Expand a statement or all of the statements on the statement list
 - EXPLAIN a statement or statements and optionally gather statistics
 - Edit a statement to create multiple versions.
 - EXPLAIN the various versions and perform a side-by-side comparison to determine the optimum form for the statement
 - Enter or change values for host variables referenced by the statement
 - Review a summary panel of the statement's execution
 - Review recommendations for the statement
-

Finding further information

Explanations of each of the fields that comprise the Statements panel and details about the use of the Statements panel can be found in “Statements” in the *!DB/EXPLAIN Reference*.

Displaying !DB/EXPLAIN Information

Overview

This unit addresses ways you can display the information !DB/EXPLAIN maintains.

Displaying objects from the Primary Menu

!DB/EXPLAIN's Primary Menu provides you with access to lists of objects that !DB/EXPLAIN manipulates:

- Plans
- DBRMs
- Packages
- Collections
- Tables
- Libraries
- Costs
- Exceptions

The object list panels provide extensive information about the objects themselves. Once you have accessed an object list panel, you can use various commands and selects to perform such tasks as:

- Binding or rebinding the object
 - Generating EXPLAINS
 - Displaying historical information
 - Displaying comparisons
 - Displaying costs
-

Displaying Housekeeping panels

All of !DB/EXPLAIN's Housekeeping panels are available as numbered selections from the Housekeeping Menu. You access the Housekeeping menu from the !DB/EXPLAIN Administration Menu or by typing **HOUS** on the command line of a !DB/EXPLAIN panel. To display a specific Housekeeping panel, enter its number in the Options field of the Housekeeping Menu.

When you change various options on !DB/EXPLAIN's Housekeeping panels, the changes remain in effect for the session in which you are working. If you have appropriate authorization, you can choose to make your changes permanent.

Displaying other administrative options

The Administration Menu provides you with access to !DB/EXPLAIN's administrative functions. In addition to Housekeeping functions, you can perform such tasks as:

- Reviewing !DB/EXPLAIN's activity log
- Reviewing and, if you are appropriately authorized, updating authorization information
- Reviewing a list of current extract IDs or extract history
- Rebuilding the object list panels generated from the current extract

Comparing Information About Objects

Overview

This unit describes the facilities !DB/EXPLAIN provides to permit you to perform comparisons.

Types of comparisons you can perform

Using !DB/EXPLAIN facilities, you can compare:

- Information about objects
 - Information about access paths
-

Why should you use the compare function?

You often spend a great deal of time and effort in developing SQL that performs within desired parameters of response time and resource usage. That time can be spent in designing a query or application, or, post-implementation, in using various DB2 features to improve performance. This activity can take the form of:

- Redesigning a database
- Adding an index
- Manipulating statistics to influence DB2's access path selection

Over time, enhancement requests result in program and database changes. New features are added to the application and new tables and indexes are added to the database. Maintenance is added to your DB2 subsystem or you upgrade to a new DB2 release. Any of these can result in changes to the access path when the plan is next bound.

Using !DB/EXPLAIN's BIND compare feature, you can determine the impact of any of these changes before BINDing your plan to the DB2 subsystem. You can perform this comparison at the plan, DBRM, package, or table level. !DB/EXPLAIN permits you to perform the comparison either interactively or in batch mode.

Comparing object information

Using the IMPACT command or the J select on the Plans, Packages, or DBRMS panel, you can compare two objects and display the results on the Compare History panel. !DB/EXPLAIN displays such information as bind date and time; creator, bound-by, and qualifier information; and object size. If you want to compare access path information about the two objects you are working with, you can do this from either the Compare History panel or from the object list panel as described next.

Comparing access path information

Once you have EXPLAINed an object, the EXPLAIN information is retained for historical purposes. You can specify the number of generations of EXPLAINs you want to retain. Using the CEXPL command from the Compare History panel, you can compare access path and other information about the current EXPLAIN with its historical counterpart.

Generating BIND commands based on comparisons

Using the COMPARE command or C select from the Plans, Packages, or DBRMS panel, you can compare EXPLAINs and, depending on the results of the comparison, generate and execute BIND or REBIND commands according to criteria you specify on Housekeeping's Bind Compare Options panel.

Reviewing Exception Conditions

Overview

This unit describes the way in which !DB/EXPLAIN identifies and displays exception conditions.

Background about exceptions

The Exceptions panel lists exceptions that exist for SQL statements. The panel shows statements that are in error and also identifies statements that can potentially affect DB2 performance. The exceptions that !DB/EXPLAIN displays are controlled by values you specify on Housekeeping's Exception Options panel and Recommendations panels.

Viewing exceptions

You can view the Exceptions panel by entering the XCPT command or X select from the following panels:

- DBRMS
- Packages
- Plans
- Statements
- Tables

You can also access the Exceptions panel directly from the !DB/EXPLAIN Primary or Sessions Menu.

Processing you can perform from the Exceptions panel

Using the selects and commands available from the exceptions panel, you can display the following types of information:

- Cost information
- The latest EXPLAIN
- SQL statements that tripped the exception
- SQL statements that contain errors

Changing Values and Playing What-if

Overview

This unit describes the use of What-if and Estimator to view and update DB2 catalog statistics.

Background about using What-if

The What-if facility allows you to analyze the effect of different catalog statistics on access path selection. You can use What-if to temporarily or permanently modify the DB2 catalog and run an EXPLAIN using these new values. The data is available in two formats. Format 1 displays the data related to access path selection. Format 2 shows greater detail from the following catalog tables: SYSTABLESPACE, SYSTABLEPART, SYSTABLES, SYSINDEXES, SYSINDEXPART, SYSTABSTATS and SYSINDEXSTATS.

Background about using Estimator

The Estimator facility permits you to update the DB2 catalog to observe how increasing or decreasing the size of tables, partitions, or indexes affects various DB2 catalog statistics. The catalog data is available in two formats. Format 1 displays the data collected from RUNSTATS. Format 2 displays data from these catalog tables: SYSTABLESPACE, SYSTABLEPART, SYSTABLES, SYSINDEXES, SYSINDEXPART, SYSTABSTATS, and SYSINDEXSTATS.

Updating a remote catalog

In the case of both What-if and Estimator, if the DB2 release is at least V2R3 you can update statistics at a remote DB2. This permits you to, for example, update the catalog on your test system to match that of your production environment. This ensures that testing of new or revised applications yields results that closely approximate the results that will occur in production.

Using !DB/EXPLAIN Facilities to Generate JCL for Common Tasks

Overview

!DB/EXPLAIN supplies various facilities to assist you in generating the JCL to run many common tasks in batch. This unit describes the ways in which you can make use of these facilities.

The !DB/EXPLAIN Online Menu

!DB/EXPLAIN provides an option from the !DB/Tools Product Selection Menu that permits you to access batch facilities using an online menu. As an option on that menu, you can request that JCL for the task you want to perform be automatically generated and either edited, stored for future use, or immediately executed. Choosing to automatically generate the JCL for the task you want to perform causes !DB/EXPLAIN to display the JCL Generation Menu. This menu, and the dialogs you can access from it, assist you in creating JCL tailored to your particular tasks and requirements. Information about using !DB/EXPLAIN's Online Menu is found in the chapter "Using the Functions of the !DB/EXPLAIN Online Menu" on page 119.

The JCL command and the JCL menu option

Another way to access the JCL Generation Menu dialogs is to select the JCL option from !DB/EXPLAIN's Primary Menu, or to enter the JCL? command on the command line of a !DB/EXPLAIN panel.

Usage Scenarios

As part of the JCL generation process, !DB/EXPLAIN provides a large number of usage scenarios from which you can select the task you want to perform. When you choose a specific scenario, the JCL to accomplish that scenario is automatically generated for you. As with other JCL generation options, you can choose to edit the generated JCL, store it for future use, or execute it immediately.

Binding or Rebinding Objects

Overview

This unit provides an overview of the task of binding or rebinding !DB/EXPLAIN objects.

Objects that can be bound

Using !DB/EXPLAIN facilities, you can issue the commands to bind these objects:

- DBRMs (as packages)
- Packages (also can be rebound)
- Plans (also can be rebound)

Panels to support bind processing

This table identifies the bind-related tasks you can perform using !DB/EXPLAIN facilities and the panels to use.

Activity You Perform	Result
Complete appropriate fields on the Package Bind panel.	!DB/EXPLAIN generates the BIND or REBIND commands for a package or the BIND commands to bind a DBRM into a package. You can optionally specify that !DB/EXPLAIN is to add or replace the package or DBRM or EXPLAIN it.
Complete appropriate fields on the Plan Bind panel.	!DB/EXPLAIN generates the BIND or REBIND commands for a plan. You can optionally specify that !DB/EXPLAIN is to add or replace the plan or EXPLAIN it.
Issue BIND command or B select from the DBRMS, Packages, or Plans panel.	!DB/EXPLAIN generates the BIND or REBIND commands for a package or plan or the BIND commands to bind a DBRM into a package.
Issue BINDADD command from the DBRMS, Packages, or Plans panel.	!DB/EXPLAIN generates the BIND or REBIND commands for a package or plan or the BIND commands to bind a DBRM into a package with the “ADD” option.

Panels to support bind processing (continued)

Activity You Perform	Result
Issue BINDCNV command or Z select from the DBRMS or Plans panel.	!DB/EXPLAIN converts the DBRMs to packages and generates the BIND commands to bind the relevant plans.
Issue BINDEXP command from the DBRMS, Packages, or Plans panel.	!DB/EXPLAIN generates the BIND or REBIND commands for a package or plan or the BIND commands to bind a DBRM into a package and EXPLAINS the object.
Issue BINDREP command from the DBRMS, Packages, or Plans panel.	!DB/EXPLAIN generates the BIND or REBIND commands for a package or the BIND commands to bind a DBRM into a package with the “REPLACE” option.
Complete appropriate fields on the Package Bind Overrides panel.	!DB/EXPLAIN uses these values to override attributes of an existing package when generating BIND or REBIND commands or make global changes to bind parameters for more than one package
Complete appropriate fields on the Plan Bind Overrides panel.	!DB/EXPLAIN uses these values to make global changes to bind parameters for more than one plan.

Other Basic Functions You Can Perform

Overview

This unit points you to information about additional basic functions you can perform when using !DB/EXPLAIN. The unit does not elaborate on these functions here, but points you to chapters in this Guide that describe these functions in detail.

Expanded information about basic functions

Review this chart to locate expanded information about the miscellaneous basic activities you can perform using !DB/EXPLAIN.

Task	Source of Additional Information
Controlling display formats	“Viewing Alternate Formats” on page 163 “Controlling the Format of the EXPLAIN Display” on page 303
Filtering	“Using Filters” on page 147 “!DB/EXPLAIN Sorts and Filters” on page 423
Sorting	“Using Sorts” on page 141 “!DB/EXPLAIN Sorts and Filters” on page 423
Using !DB/EXPLAIN commands	“Using Commands” on page 127 “Commands Available from Object List Panels” on page 387 “Dictionary of !DB/EXPLAIN Commands” on page 393
Using !DB/EXPLAIN selects	“Selects Available from Object List Panels” on page 409 “Dictionary of Selects” on page 411
Using ISPF facilities	“Using ISPF Facilities within !DB/EXPLAIN” on page 171

Introducing Fundamentals

Chapter 4.

Accessing and Exiting !DB/EXPLAIN

Introduction

This chapter describes the procedures for accessing and exiting !DB/EXPLAIN.

This chapter assumes that !DB/EXPLAIN is installed. If !DB/EXPLAIN is not yet installed, see the *!DB/Tools Installation and Customization Guide*. When you access !DB/EXPLAIN, the DB2 LOADLIB (DSNLOAD) is assumed to be in your system LINKLIST. See “Before You Begin” in the *!DB/Tools Installation and Customization Guide* for more information.

Chapter contents

Accessing !DB/EXPLAIN	70
Access from the !DB/Tools Product Selection Menu	71
Access by Invoking the !DB/EXPLAIN CLIST	73
Access from an Edit Macro	77
Access Using the Edit Macro KTEXPL	78
Initiating an EXPLAIN using KTEXPL	81
Access Using the Edit Macro KTEXPLB	86
Access Using the EXEC KTEQMF	87
Access Using the EXEC KTEON	90
Access from EXEC KTEXPLA	92
Access from EXEC KTEXPLBA	95
Access from EXEC KTEXPLD	98
Determining Whether to Display Statements	101
Access from Candle's OMEGAMON II for DB2 (First Access)	104
Accessing !DB/EXPLAIN Batch Facilities	105
Exiting !DB/EXPLAIN	106

Accessing !DB/EXPLAIN

Overview

This unit provides information about accessing !DB/EXPLAIN.

Background about accessing !DB/EXPLAIN

There are several different ways of accessing !DB/EXPLAIN. Each way has its advantages and gives you more flexibility in using !DB/EXPLAIN. Once inside !DB/EXPLAIN from any access, you can navigate the product fully.

Methods to access !DB/EXPLAIN

You can access !DB/EXPLAIN:

- From the !DB/Tools Product Selection Menu
- By invoking the !DB/EXPLAIN EXEC KTE
- From an edit session by invoking the !DB/EXPLAIN edit macros KTEXPL or KTEXPLB
- By invoking the EXEC KTEQMF from within QMF to EXPLAIN QMF SQL queries, QMF prompted queries, and QMF query-by-example (QBE) queries
- From the ISPF command line using the KTEON EXEC
- From the EXECs KTEXPLA, KTEXPLBA, and KTEXPLD
- From Candle's OMEGAMON II for DB2
- Through batch jobs

Access from the !DB/Tools Product Selection Menu

Overview

This unit describes the procedure for accessing !DB/EXPLAIN from the !DB/Tools Product Selection Menu.

Procedure to access !DB/EXPLAIN from the !DB/Tools Product Selection Menu

Follow these instructions to access !DB/EXPLAIN from the !DB/Tools Product Selection Menu.

Step	Action
1	Access Candle !DB/Tools by typing the appropriate value on the command line: <ul style="list-style-type: none"> ● If you are on the ISPF Option 6 (Command) panel, type %KTC. ● If you are on any other ISPF panel, type TSO %KTC. <p>Note: Use of “%” is optional, but you can realize a performance improvement when using % in front of a CLIST name.</p>
2	Press Enter. Result: The !DB/Tools Product Selection Menu displays.
3	You can access !DB/EXPLAIN in one of several ways: <ul style="list-style-type: none"> ● To access !DB/EXPLAIN through the !DB/EXPLAIN Primary Menu, type 4 in the Select Product ID field. ● To access !DB/EXPLAIN through the KTEON EXEC, type 5 in the Select Product ID field.
4	Ensure that the extract ID in the Select Extract ID field of the !DB/Tools Product Selection Menu is the one you want. If the extract ID is not the one you want, or you don't know what extract IDs are available, see “Selecting the extract ID” on page 72.
5	Press Enter. Result: You receive either the !DB/EXPLAIN Primary Menu or the !DB/EXPLAIN KTEON EXEC menu, depending on the value you entered in Step 3.

Selecting the extract ID

Follow this procedure to insert or override the extract ID on the !DB/Tools Product Selection Menu.

Step	Action
1	Review the extract ID in the Select Extract ID field of the !DB/Tools Product Selection Menu.
2	To override the displayed extract ID, overtype it with the extract ID you want and press Enter.
3	If you do not know what extract ID you want or what extract IDs are available, type ? in the Extract ID field and press Enter. Result: The !DB/Tools—DB2 Subsystem Name Table panel displays. On it, you can select the extract ID you want by typing a character in that extract ID's selection column.

Access by Invoking the !DB/EXPLAIN CLIST

Overview

To access !DB/EXPLAIN directly without going through the !DB/Tools Product Selection Menu, use the !DB/EXPLAIN CLIST.

Procedure to access !DB/EXPLAIN from the !DB/EXPLAIN CLIST

Follow these instructions to access !DB/EXPLAIN from the !DB/EXPLAIN CLIST.

Step	Action
1	<p>Access the CLIST by typing the appropriate value on the command line:</p> <ul style="list-style-type: none"> ● If you are on the ISPF Option 6 (Command) panel, type %KTE <i>parmlist</i> ● If you are on any other ISPF panel, type TSO %KTE <i>parmlist</i> <p><i>parmlist</i> represents the optional parameters that can be included in the command and are described in “Optional parameters” on page 74</p> <p>Note: Use of “%” is optional, but you can realize a performance improvement when using % in front of a CLIST name.</p>
2	<p>Press Enter.</p> <p>Result: The !DB/EXPLAIN Primary Menu displays.</p> <p>The value shown in the Extract ID field is taken from your ISPF profile unless you have specified an extract ID in your input parameter list or you are executing !DB/EXPLAIN for the first time.</p> <p>If you are executing !DB/EXPLAIN for the first time, see “If this is your first execution of !DB/EXPLAIN” on page 76 for information about how the extract ID is assigned.</p>

Optional parameters

The table below describes the optional parameters, any or all of which may be included following **%KTE** in the command used to invoke the CLIST. Examples of the use of these parameters are found in “Command examples” on page 75.

IF a value for this parameter is specified...	THEN...
extract_id	The named extract is referenced in this session; for example, TSO %KTE D31A references the extract D31A .
SYSPROF(dsname)	Specifying the name of a System Profile Dataset for this parameter overrides the value for the System Profile Dataset named in your ISPF profile or the KTCSETUP CLIST.
USERPROF(dsname)	Specifying the name of a User Profile Dataset for this parameter overrides the value for the User Profile Dataset named in your ISPF profile or the KTCSETUP CLIST.
AUXPROF(dsname)	Specifying the name of an Auxiliary (sometimes called Group) Profile Dataset for this parameter overrides the value for the Auxiliary Profile Dataset named in your ISPF profile or the KTCSETUP CLIST.
REFRESH(Y)	Specifying REFRESH(Y) refreshes the ISPF Profile values. You should perform a REFRESH when changing dataset names or when switching between profile datasets, in order to clear stored values.

Command examples

Valid examples using the !DB/EXPLAIN CLIST and optional commands appear below. Dataset names are in the standard ISPF format, either fully qualified within quotes or implicitly prefixed.

Example	Explanation
----------------	--------------------

TSO %KTE DB2	
---------------------	--

This example references the extract ID **DB2**.

TSO %KTE USERPROF(mydata.profile)	
--	--

!DB/EXPLAIN uses the default extract ID and the User Profile Dataset named **mydata.profile** overrides the value in your ISPF profile or KTCSETUP CLIST.

TSO %KTE SYSPROF('tdtda9.mydata.profile')	
--	--

!DB/EXPLAIN uses the default extract ID. The System Profile Data Set named **mydata.profile** overrides the value in your ISPF profile or KTCSETUP CLIST. In this case, the fully qualified data set name has been used.

TSO %KTE SYSPROF(mydata.profile) DB2 AUXPROF(auxdata.prof)	
---	--

The datasets **mydata.profile** and **auxdata.prof** override the values in the System Profile Data Set and Auxiliary Profile dataset respectively in your ISPF profile or the CLIST KTCSETUP. Note that the extract ID, in this case, **DB2**, can be placed anywhere in the input command.

If this is your first execution of !DB/EXPLAIN

If you leave the extract ID parameter blank or specify an invalid extract ID, *and* this is your first execution of !DB/EXPLAIN, follow these steps.

Step	Action
1	When you enter !DB/EXPLAIN, !DB/EXPLAIN displays the Primary Menu panel with the value NONE in the extract ID field.
2	On the Primary Menu, type ? in the extract ID field.
3	Press Enter. Result: !DB/EXPLAIN displays a list of valid extract IDs.
4	Select a valid extract ID. Result: !DB/EXPLAIN returns to the Primary Menu, placing the selected ID in the Extract ID field of the Primary Menu and also into your ISPF profile.
5	On subsequent executions, !DB/EXPLAIN uses the extract ID from your ISPF profile unless you specifically override this choice by using the <i>extract_id</i> parameter on the !DB/EXPLAIN CLIST.

Access from an Edit Macro

Overview

To interactively access !DB/EXPLAIN during the application development process, either before or after your code has been bound, you can use one of the supplied edit macros. This unit tells you where to find information about the edit macros.

Initiating an edit macro

You can initiate these macros from the ISPF command line, or you can initiate them from another editor running under ISPF.

Sources of information

The chart below tells you where to find information on accessing !DB/EXPLAIN from the edit macros distributed with !DB/EXPLAIN.

Edit Macro	Information Source
KTEXPL: EXPLAIN SQL code	“Access Using the Edit Macro KTEXPL” on page 78
KTEXPLB: EXPLAIN the SQL statements associated with the BIND commands in a member	“Access Using the Edit Macro KTEXPLB” on page 86

Access Using the Edit Macro KTEXPL

Overview

This unit describes how to interactively access !DB/EXPLAIN function using the edit macro KTEXPL.

Use of KTEXPL

KTEXPL is typically used by application developers to EXPLAIN their SQL code during program development, either before or after statements are bound into plans or packages.

Defining a default format for your input file

SQL code you want to EXPLAIN can be imbedded in assembler, C, COBOL, Fortran, PL/I, or another format you specify, for example, SAS. !DB/EXPLAIN has provided some default formats on the SQL Formats for KTEXPL panel. You can assign any letter to any format, but you *must* ensure that the format descriptors that follow the format on the same line agree with the format type you define.

If you often use the same file format for the file you process using KTEXPL, you can save time by defining that format as the default.

Defining a default format for your input file (continued)

This table describes how to define a default format for your input file.

Step	Action
1	On the !DB/EXPLAIN Housekeeping Menu, select SQL Parameters for KTEXPL.
2	<p>Press Enter.</p> <p>Result: The SQL Formats for KTEXPL panel displays.</p> <div data-bbox="573 611 1248 1056" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==>> S Q L F O R M A T S F O R K T E X P L PERMANENT? ==>> (Y Yes N No) Format ==>> Format of the Host Comment SQL All Dash Input File LHS RHS Col Flag Col Del Del Delimiter SQL Host? Type ----- A - ASSEMBLER 1 71 72 : 1 * NA NA N N B - C 1 72 0 : 0 /* */ ; N N D C - COBOL 8 72 0 : 7 * NA END-EXEC N Y F - FORTRAN 7 72 6 : 1 * NA NA N N P - PLI 2 72 0 : 0 /* */ ; N N Q - QMF 1 79 0 & 0 -- NA ; Y N S - SPUFI 1 71 0 : 0 -- NA ; Y N T - UNDEFINED 0 0 0 : 0 U - UNDEFINED 0 0 0 : 0 V - UNDEFINED 0 0 0 : 0 W - UNDEFINED 0 0 0 : 0 X - UNDEFINED 0 0 0 : 0 ENTER to process END to cancel </pre> </div>
3	In the Format field, type the letter that identifies the default format of the input file.
4	To save that format selection as the default, type Y in the Permanent field.
5	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN saves the format you selected as the default format and bypasses the display of the SQL Formats for KTEXPL panel on subsequent executions.</p>

Determining whether to display statements

You have two choices when using KTEXPL to access !DB/EXPLAIN:

1. You can choose to see the Statements panel first. This panel contains the EXPLAINable statements in your input file. You can then choose to EXPLAIN any or all of them.
2. You can choose to bypass the Statements display and go directly to the EXPLAIN panel. Doing so provides you with an EXPLAIN of all of the EXPLAINable SQL statements in your input file.

Information about how to display Statements or bypass their display is found in “Determining Whether to Display Statements” on page 101.

Initiating an EXPLAIN using KTEXPL

Overview

This unit describes the steps to initiate an EXPLAIN using the edit macro KTEXPL.

Variations in using KTEXPL

There are several variations on the procedure. The table below describes them.

Task you want to perform	See the following variation
You want to initiate an EXPLAIN using KTEXPL, you have previously defined a default format, and you want to use that format.	“Procedure when you have defined a default format” on page 82
You want to initiate an EXPLAIN using KTEXPL, you have previously defined a default format, and you want to override the format you previously specified.	“Procedure to override a defined default format” on page 83
You want to initiate an EXPLAIN using KTEXPL, you have <i>not</i> previously defined a default format, and you <i>do not</i> know the format code.	“Procedure when there is no default and you do not know the format code” on page 84
You want to initiate an EXPLAIN using KTEXPL, you have <i>not</i> previously defined a default format, but you <i>do</i> know the format code	“Procedure when there is no default but you know the format code” on page 85

Procedure when you have defined a default format

When you have previously defined a default format for your input file and want to use that format when executing KTEXPL, follow this procedure.

Step	Action
1	In an edit session, open the member you want to EXPLAIN.
2	Use any valid block copy command to block off the SQL code. If you do not use the block command, the entire member will be scanned.
3	<p>If you want to use the default format you have specified previously on the SQL Formats for KTEXPL panel, type the following on the command line:</p> <p style="text-align: center;">KTEXPL <i>extract_id</i></p>
4	<p>Press Enter.</p> <p>Result: Depending on whether or not you specified bypassing the Statements display, one of the following occurs:</p> <ul style="list-style-type: none"> ● If you have specified bypassing the Statements display, !DB/EXPLAIN analyzes the SQL statements in your file and displays the EXPLAIN panel in the format you specified. ● If you have not specified bypassing the Statements display, !DB/EXPLAIN displays the Statements panel. <p>Processing occurs as described in “Determining Whether to Display Statements” on page 101. Then the EXPLAIN panel displays.</p>

Procedure to override a defined default format

To override a default format you have previously defined, follow this procedure.

Step	Action
1	In an edit session, open the member you want to EXPLAIN.
2	Use any valid block copy command to block off the SQL code. If you do not use the block command, the entire member will be scanned.
3	<p>To override the previously specified default format for this session, type the following on the command line:</p> <p style="text-align: center;">KTEXPL <i>extract_id format</i></p>
4	<p>Press Enter.</p> <p>Result: Depending on whether or not you specified bypassing the Statements display, one of the following occurs:</p> <ul style="list-style-type: none"> ● If you have specified bypassing the Statements display, !DB/EXPLAIN analyzes the SQL statements in your file and displays the EXPLAIN panel in the format you specified. ● If you have not specified bypassing the Statements display, !DB/EXPLAIN displays the Statements panel. <p>Processing occurs as described in “Determining Whether to Display Statements” on page 101. Then the EXPLAIN panel displays.</p>

Procedure when there is no default and you do not know the format code

If you have not defined a default format and don't know the applicable format codes, follow this procedure.

Step	Action
1	In an edit session, open the member you want to EXPLAIN.
2	Use any valid block copy command to block off the SQL code. If you do not use the block command, the entire member will be scanned.
3	If you have not previously specified a default format and do <i>not</i> know the format code, type the following on the command line: KTEXPL <i>extract_id</i>
4	Press Enter. Result: The SQL Formats for KTEXPL panel displays. Enter a format code and whether or not you want the format code to be permanent.
5	Press Enter. Result: Depending on whether or not you specified bypassing the Statements display, one of the following occurs: <ul style="list-style-type: none"> ● If you have specified bypassing the Statements display, !DB/EXPLAIN analyzes the SQL statements in your file and displays the EXPLAIN panel in the format you specified. ● If you have not specified bypassing the Statements display, !DB/EXPLAIN displays the Statements panel. Processing occurs as described in “Determining Whether to Display Statements” on page 101. Then the EXPLAIN panel displays.

Procedure when there is no default but you know the format code

Use this procedure when you have not specified a default value but know the format code you want to use.

Step	Action
1	In an edit session, open the member you want to EXPLAIN.
2	Use any valid block copy command to block off the SQL code. If you do not use the block command, the entire member will be scanned.
3	<p>If you have not specified a default format but know the format code, type the following on the command line:</p> <p style="text-align: center;">KTEXPL <i>extract_id format</i></p>
4	<p>Press Enter.</p> <p>Result: Depending on whether or not you specified bypassing the Statements display, one of the following occurs:</p> <ul style="list-style-type: none"> ● If you have specified bypassing the Statements display, !DB/EXPLAIN analyzes the SQL statements in your file and displays the EXPLAIN panel in the format you specified. ● If you have not specified bypassing the Statements display, !DB/EXPLAIN displays the Statements panel. <p>Processing occurs as described in “Determining Whether to Display Statements” on page 101. Then the EXPLAIN panel displays.</p>

Saved values for extract ID and format parameters

When you invoke KTEXPL specifying the extract ID and format, the values for these parameters are stored in your ISPF profile and reused for the next invocation of KTEXPL. If you do not specify valid values for the extract ID and format during invocation, or if the values read from your ISPF profile are invalid, !DB/EXPLAIN displays the SQL Formats for KTEXPL panel to show you the available choices and permit you to make a selection.

Access Using the Edit Macro KTEXPLB

Overview

This unit describes the procedure to interactively access !DB/EXPLAIN using the edit macro KTEXPLB.

Use of KTEXPLB

Invoking KTEXPLB in an edit session generates an EXPLAIN of the SQL statements associated with the BIND commands in the member to obtain access path information.

Search order for DBRMs specified in BIND commands

When you execute KTEXPLB, the dataset or datasets specified by the allocated DBRMLIB DD statement are searched for the DBRMs specified in the BIND commands. If the DBRMs are not found in the datasets specified in the allocated DBRMLIB DD statement, the datasets named in the LIBRARY() parameter for the BIND commands or the DBRMLIB DD of the BIND command's JCL are searched next.

Procedure to use KTEXPLB

Follow this procedure to use edit macro KTEXPLB.

Step	Action
1	In an ISPF edit session, open the member you want to EXPLAIN.
2	Type KTEXPLB on the command line.
3	<p>Press Enter.</p> <p>Result: One of the following occurs:</p> <ul style="list-style-type: none"> • If you have specified that the Statements panel should be displayed, !DB/EXPLAIN displays it now. • If you have specified bypassing the Statements panel, !DB/EXPLAIN displays the EXPLAIN panel containing access path information for the BIND commands in your edit member. <p>Processing occurs as described in “Determining Whether to Display Statements” on page 101.</p>

Access Using the EXEC KTEQMF

Overview

This unit describes the procedure to use the QMF-to-!DB/EXPLAIN bridge.

Use of the bridge

The KTEQMF bridge allows you to EXPLAIN QMF queries and procedures directly from QMF. These queries may be SQL queries, prompted queries, or query-by-example (QBE) queries. You interactively access !DB/EXPLAIN from a QMF query or procedure using the EXEC KTEQMF.

Invoking KTEQMF

Invoking KTEQMF while in a QMF session generates an EXPLAIN of your query or, for a QMF procedure, all of the queries it executes. You can use KTEQMF from the command line of your query or procedure. Alternatively, you can set up a command synonym (essentially an “alias” for KTEQMF) in the QMF COMMAND_SYNONYMS control table. Once you have established this command synonym, it can be entered from the command line of your query or procedure in place of entering KTEQMF.

Using the German language version of the QMF bridge

If you are running QMF in German and want to use the QMF to !DB/EXPLAIN bridge, substitute the name KTEQMFGE for KTEQMF wherever it appears in the instructions in this unit.

Rule for using the QMF bridge

In order to use the QMF bridge, the version of QMF that you are running must be compatible with the DB2 subsystem and version and the !DB/EXPLAIN extract ID that you have entered on the !DB/Tools DB2-Specific Information panel.

Procedure for using KTEQMF from the command line

Follow this procedure to use edit macro KTEQMF.

Step	Action
1	In a QMF session, access the query or procedure you want to EXPLAIN.
2	Type TSO %KTEQMF <i>extract_id</i> on the command line. If you are running the German language version, type TSO %KTEQMFGE <i>extract_id</i> on the command line.
3	Press Enter. Result: One of the following occurs: <ul style="list-style-type: none"> • If you have specified that the Statements panel should be displayed, !DB/EXPLAIN displays it now. • If you have specified bypassing the Statements panel, !DB/EXPLAIN displays the EXPLAIN of the SQL statements in your query or in all of the queries called by your procedure. Processing occurs as described in “Determining Whether to Display Statements” on page 101.

Procedure for setting up a QMF command synonym

To set up a QMF command synonym, you must insert a row into QMF's applicable COMMAND_SYNONYMS control table. (QMF's Q.PROFILES table controls which COMMAND_SYNONYMS table you are associated with.) Use the commands below to insert the row.

```
INSERT INTO Q.COMMAND_SYNONYMS
(VERB,OBJECT,SYNONYM_DEFINITION,REMARKS)
VALUES ('sssssss',NULL,'TSO %KTEQMF &ALL', '!DB/EXPLAIN BRIDGE')
```

sssssss represents the command synonym you want to use to invoke the QMF bridge. You can choose any value for this synonym except those that begin with the first three letters of any QMF command. Thus, the letters “EXP.” are not acceptable since they are the first three letters of the EXPORT command.

If you are running the German language version, the third value parameter should be entered as '**TSO %KTEQMFGE &ALL**'.

Procedure for using a QMF command synonym

Once you have set up a command synonym, you can use it by following the steps in this table.

Step	Action
1	In a QMF session, access the query or procedure you want to EXPLAIN.
2	Type <i>command_synonym extract_id</i> on the command line. Example: If you defined a command synonym EXQMF and want to reference extract ID D31A, type the following on the command line of your query or procedure: EXQMF D31A
3	Press Enter. Result: !DB/EXPLAIN displays the EXPLAIN of the SQL statement in your query or in all of the queries called by your procedure.

Access Using the EXEC KTEON

Overview

This unit tells you how to access !DB/EXPLAIN using the EXEC KTEON.

Background about using KTEON

You can use the EXEC KTEON to interactively access !DB/EXPLAIN function during an ISPF session. Your interface to KTEON is via the !DB/EXPLAIN Online Menu. KTEON invokes the EXECs KTEXPLA, KTEXPLBA, and KTEXPLD. You can also invoke these EXECs directly while in an ISPF session.

Invoking the KTEXPL* EXECs directly from ISPF

The chart below tells you where to find information on accessing !DB/EXPLAIN directly from the ISPF command line using CLISTs distributed with !DB/EXPLAIN.

EXEC	Information Source
KTEXPLA: EXPLAIN SQL statements	“Access from EXEC KTEXPLA” on page 92
KTEXPLBA: EXPLAIN SQL statements associated with BIND commands	“Access from EXEC KTEXPLBA” on page 95
KTEXPLD: EXPLAIN DBRM members	“Access from EXEC KTEXPLD” on page 98

Functions supported by the Online Menu

A complete discussion of the functions supported by the Online Menu, KTEON, is found in “Using the Functions of the !DB/EXPLAIN Online Menu” on page 119 in the chapter “Accessing !DB/EXPLAIN Functions.”

Scenarios illustrating the use of KTEON

Scenarios illustrating ways you can use the KTEON EXEC for online and batch executions are found in “Accessing !DB/EXPLAIN Using the KTEON EXEC” on page 240 and “Accessing !DB/EXPLAIN Using the KTEON EXEC to Generate JCL” on page 244 respectively.

Procedure to access !DB/EXPLAIN using KTEON

To access !DB/EXPLAIN using KTEON, follow the procedure in this table.

Step	Action
1	<p>To access KTEON, use one of these methods:</p> <ul style="list-style-type: none"> ● On the !DB/Tools Product Selection Menu, choose Option 5, !DB/EXPLAIN Online Menu, or ● On the command line of any ISPF panel, enter TSO KTEON. You can optionally include the extract ID.
2	<p>Press Enter.</p> <p>Result: The Online Menu panel is presented on which you can select various objects to EXPLAIN.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> O N L I N E M E N U Extract ID ==> D31A Estae Flag ==> Y Create Batch JCL? ==> N (Y Yes N No) Specify Additional Data Sets? ==> N (Y Yes N No) Specify Member Lists? ==> N (Y Yes N No) Option ==> 1 1 SQL Statements Data Set Name : TDDB00.USER.SQL(SQL) File Format : S 2 BIND Commands Data Set Name : TDDB00.USER.BIND Allocate DBRMLIB?: N (Y Yes N No E Edit DBRMLIB concatenation) 3 DBRM Data Set Name : TDDB00.USER.DBRM Owner/Qualifier : TDDB00 Plan: ----- (C) Copyright CANDLE Corp. 1993 - 1999 ----- </pre> </div>
3	<p>Proceed as described in “Using the Functions of the !DB/EXPLAIN Online Menu” on page 119.</p>

Access from EXEC KTEXPLA

Overview

This unit provides the syntax for accessing !DB/EXPLAIN directly from EXEC KTEXPLA.

Use of KTEXPLA

KTEXPLA is used to EXPLAIN all SQL statements in a data set.

Syntax

The syntax for KTEXPLA is:

KTEXPLA *dsn* DB2(*extract_id*) FORMAT(*f*)

where:

- *dsn* is the name of the data set that contains the statements to be EXPLAINed. The data set name can be physical sequential or partitioned. If partitioned, you can specify a member name, a member name pattern mask, or an asterisk, or you can omit the member name. If you do not specify a member name, or you enter a member name pattern mask, !DB/EXPLAIN presents you with a list of members from which you can select the members to be processed. If you enter an asterisk, !DB/EXPLAIN immediately processes all members.
- *extract_id* is the extract ID to be referenced
- *f* is the format (for example, *A* for assembler, *C* for C, *F* for FORTRAN) of the data set

Procedure for using KTEXPLA

Use the chart below to access !DB/EXPLAIN from EXEC KTEXPLA. Remember that, in steps 3 and 4 below, if the data set you are working with is a partitioned data set, the field (**mymember**) can be one of the following:

- A member name
!DB/EXPLAIN EXPLAINs the SQL statements in that member
- Asterisk (*)
!DB/EXPLAIN immediately processes all members.
- Blank
!DB/EXPLAIN presents the entire list of members from which you can select the members to be EXPLAINed.
- A member name pattern mask
!DB/EXPLAIN presents a list of all members in the data set matching the pattern. You can then select those to be EXPLAINed.

Step	Action
1	In your ISPF session, select Option 3—Utilities or Option 6—Commands the ISPF Primary Options Menu.
2	Press Enter. <ul style="list-style-type: none"> ● If you selected Option 3, go to Step 3. ● If you selected Option 6, go to Step 4.
3	<p>If you entered Option 3, the Utility Selection Menu displays.</p> <ul style="list-style-type: none"> ● Select Option 4 to obtain the Data Set List Utility panel. ● On that panel, in the field Dsname Level, enter the data set name and press Enter. Example: DSNAME LEVEL===>TDDDB.DE2.ASM ● When the data set list appears, type the following on the the line containing the data set name: KTEXPLA /(mymember) DB2(D31) FORMAT(A) <ul style="list-style-type: none"> ● Go to Step 5.

Procedure for using KTEXPLA (continued)

Step	Action
4	<p>If you entered Option 6, the ISPF Commands panel displays. After the ==>, type the EXEC command; for example:</p> <pre style="text-align: center;">KTEXPLA 'TDDB.DE2.ASM(mymember)' DB2(D31) FORMAT(A)</pre> <p>or</p> <pre style="text-align: center;">KTEXPLA 'TDDB.DE2.ASM(*)' DB2(D31) FORMAT(A)</pre> <p>Go to Step 5.</p>
5	<p>Press Enter.</p> <p>Result: One of the following occurs:</p> <ul style="list-style-type: none"> ● If you have specified that the Statements panel should be displayed, !DB/EXPLAIN displays it now. If you selected multiple members from a member list to be EXPLAINed, the statements from all of the selected members appear on the Statements display. ● If you have specified bypassing the Statements panel, !DB/EXPLAIN displays the EXPLAIN of the SQL statements in data set member or members whose format is assembler. <p>Processing occurs as described in “Determining Whether to Display Statements” on page 101.</p>

Access from EXEC KTEXPLBA

Overview

This unit provides the syntax for accessing !DB/EXPLAIN directly from EXEC KTEXPLBA.

Use of KTEXPLBA

KTEXPLBA is used to EXPLAIN the SQL statements associated with all BIND commands in a data set.

Search order for DBRMs specified in BIND commands

When you execute KTEXPLBA, the data set or datasets specified by the allocated DBRMLIB DD statement are searched for the DBRMs specified in the BIND commands. If the DBRMs are not found in the datasets specified in the allocated DBRMLIB DD statement, the datasets named in the LIBRARY() parameter for the BIND commands or the DBRMLIB DD of the BIND command's JCL are searched next.

Syntax

The syntax for KTEXPLBA is:

```
KTEXPLBA dsn DB2(extract_id)
```

where:

- *dsn* is the name of the data set that contains the SQL statements to be EXPLAINed. The data set name can be physical sequential or partitioned. If partitioned, you can specify a member name, a member name pattern mask, or an asterisk, or you can omit the member name. If you do not specify a member name, or you enter a member name pattern mask, !DB/EXPLAIN presents you with a list of members from which you can select the members to be processed. If you enter an asterisk, !DB/EXPLAIN immediately processes all members.
- *extract_id* is the extract ID you want to reference

Procedure to use KTEXPLBA

Use the chart below to access !DB/EXPLAIN from EXEC KTEXPLBA. Remember that, in steps 3 and 4 below, if the data set you are working with is a partitioned data set, the field (**mymember**) can be one of the following:

- A member name
!DB/EXPLAIN EXPLAINs the SQL statements in that member
- Asterisk (*)
!DB/EXPLAIN immediately processes all members.
- Blank
!DB/EXPLAIN presents the entire list of members from which you can select the members to be EXPLAINed.
- A member name pattern mask
!DB/EXPLAIN presents a list of all members in the data set matching the pattern. You can then select those to be EXPLAINed.

Step	Action
1	In your ISPF session, select Option 3—Utilities or Option 63—Commands on the ISPF Primary Options Menu.
2	Press Enter. <ul style="list-style-type: none"> ● If you selected Option 3, go to Step 3. ● If you selected Option 6, go to Step 4.
3	<p>If you entered Option 3, the Utility Selection Menu displays.</p> <ul style="list-style-type: none"> ● Select Option 4 to obtain the Data Set List Utility panel. ● On that panel, in the field Dsname Level, enter the data set name and press Enter. Example: DSNAME LEVEL===>TDDB.DE2.JCL ● When the data set list appears, enter the following on the the line containing the data set name: <div style="text-align: center;">KTEXPLBA /(mymember) DB2(D31)</div> <ul style="list-style-type: none"> ● Go to Step 5.

Procedure for using KTEXPBLA (continued)

Step	Action
4	<p>If you entered Option 6, the ISPF Commands panel displays. After the ==>, enter the EXEC command; for example:</p> <pre style="text-align: center;">KTEXPLBA 'TDDB.DE2.JCL(mymember)' DB2(D31)</pre> <p>or</p> <pre style="text-align: center;">KTEXPLBA 'TDDB.DE2.JCL(mym*b*)' DB2(D31)</pre> <p>Go to Step 5.</p>
5	<p>Press Enter.</p> <p>Result: One of the following occurs:</p> <ul style="list-style-type: none"> ● If you have specified that the Statements panel should be displayed, !DB/EXPLAIN displays it now. If you selected multiple members from a member list to be EXPLAINed, the statements from all of the selected members appear on the Statements display. ● If you have specified bypassing the Statements panel, !DB/EXPLAIN displays the EXPLAIN of the SQL statements associated with the BIND commands in data set member or members. <p>Processing occurs as described in “Determining Whether to Display Statements” on page 101.</p>

Access from EXEC KTEXPLD

Overview

This unit provides the syntax for accessing !DB/EXPLAIN directly from EXEC KTEXPLD.

Use of KTEXPLD

KTEXPLD is used to EXPLAIN all SQL statements in a DBRM.

Syntax

The syntax for KTEXPLD is:

```
KTEXPLD DSN(dsn) DBRM(member) DB2(extract_id) OWNER(qualifier) PLAN(plan_name)
```

where:

- *dsn* is the partitioned data set that contains the DBRM.
- *member* is the member name or member pattern mask for the members that contains the DBRM whose statements are to be EXPLAINed. If you do not specify DBRM(*member*), !DB/EXPLAIN displays a member list from which you can select the members to be processed. If you specify a member name pattern mask, !DB/EXPLAIN presents you with a list of qualifying members from which you can select the members to be processed. If you specify *, all members in the partitioned data set are processed.
- *extract_id* is the extract ID you want to reference
- *qualifier* is the value to use for unqualified tables contained in the DBRM
- *plan_name* is the plan name

Procedure to use KTEXPLD

Use the chart below to access !DB/EXPLAIN from EXEC KTEXPLD. Remember that, in steps 3 and 4 below, the data set you are working with is a partitioned data set. The field (**mymember**) on the parameter DBRM can be one of the following:

- A member name
!DB/EXPLAIN EXPLAINS the SQL statements in that member
- (*)—for example, DBRM(*)
!DB/EXPLAIN immediately processes all members
- blank—for example, DBRM()
!DB/EXPLAIN presents a list of all members in the data set from which you can select those to be EXPLAINED.
- A member name pattern mask
!DB/EXPLAIN presents a list of all members in the data set matching the pattern. You can then select those to be EXPLAINED.

Step	Action
1	In your ISPF session, select Option 3—Utilities or Option 6—Commands on the ISPF Primary Option Menu.
2	Press Enter. <ul style="list-style-type: none"> ● If you selected Option 3, go to Step 3. ● If you selected Option 6, go to Step 4.
3	<p>If you entered Option 3, the Utility Selection Menu displays.</p> <ul style="list-style-type: none"> ● Select Option 4 to obtain the Data Set List Utility panel. ● On that panel, in the field Dsname Level, enter the DBRM name and press Enter. Example: DSNAME LEVEL===>TDB.DE2.DBRM ● When the data set list appears, enter the following on the the line containing the DBRM member name: <pre style="margin-left: 40px;">KTEXPLD DSN(/) DBRM(mymember) DB2(D31) OWNER(TDB) PLAN(name)</pre> <p style="text-align: center;">or</p> <pre style="margin-left: 40px;">KTEXPLD DSN(/) DBRM(*memb*) DB2(D31) OWNER(TDB) PLAN(name)</pre> <ul style="list-style-type: none"> ● Go to Step 5.

Procedure to use KTEXPLD (continued)

Step	Action
4	<p>If you entered Option 6, the ISPF Commands panel displays. After the ==>, enter the EXEC command; for example:</p> <pre data-bbox="609 415 1422 443">KTEXPLD DSN('TDB.DE2.DBRM') DBRM(mymember) DB2(D31) OWNER(qualifier)</pre> <p>Go to Step 5.</p>
5	<p>Press Enter</p> <p>Result: One of the following occurs:</p> <ul data-bbox="570 598 1425 863" style="list-style-type: none"> • If you have specified that the Statements panel should be displayed, !DB/EXPLAIN displays it now. If you selected multiple members from a member list to be EXPLAINed, the statements from all of the selected members appear on the Statements display. • If you have specified bypassing the Statements panel, !DB/EXPLAIN displays the EXPLAIN of the SQL statements in DBRM member or members. Unqualified tables in the DBRM are qualified with the value specified for owner (in the example above, TDB). If a plan name is not specified, it defaults to the DBRM name. <p>Processing occurs as described in “Determining Whether to Display Statements” on page 101.</p>

Determining Whether to Display Statements

Overview

This unit helps you determine whether or not you want to display the Statements panel when performing an EXPLAIN and provides instructions to help you implement your choice.

Background about displaying statements

When you access !DB/EXPLAIN from an edit macro, CLIST, or EXEC, you have a choice as to whether or not the SQL statements in your input are displayed.

1. You can choose to see the Statements panel first. This panel contains the EXPLAINable statements in your input file. You can then choose to EXPLAIN any or all of them.
 2. You can choose to bypass the Statements display and go directly to the EXPLAIN panel. Doing so provides you with an EXPLAIN of all of the EXPLAINable SQL statements in your input file.
-

Making your choice permanent

If you want your decision about displaying statements to apply across !DB/EXPLAIN sessions, you can type **Y** in the Permanent field of the Bypass Statements Display panel. You must be appropriately authorized to permanently save your selection. If you type **Y** and are not appropriately authorized, !DB/EXPLAIN displays an error message. If you receive an error message, type **N** in the Permanent field and continue with your work.

If you are appropriately authorized to make your choice permanent, and you have selected the **Y** option, !DB/EXPLAIN will display the Statements panel each time you execute your edit macro.

Procedure to view the Statements display

To ensure that the Statements panel is displayed during each execution of the edit macro you are using, follow this procedure.

Step	Action
1	On the !DB/EXPLAIN Administration Menu, select Housekeeping, or type HOUSE on the command line of a !DB/EXPLAIN panel to access the Housekeeping Menu.
2	On the !DB/EXPLAIN Housekeeping Menu, select SQL Defaults.
3	Press Enter. Result: The SQL Defaults panel displays.
4	On the SQL Defaults panel, type N in the Bypass Statements Display field.
5	Press Enter. Result: !DB/EXPLAIN displays the Statements panel. If you made this option permanent, !DB/EXPLAIN displays the Statements panel for subsequent executions.

Procedure to bypass the Statements display

To bypass the Statements display, follow the procedure below.

Step	Action
1	On the !DB/EXPLAIN Administration Menu, select Housekeeping, or type HOURSE on the command line of a !DB/EXPLAIN panel to access the Housekeeping Menu.
2	On the !DB/EXPLAIN Housekeeping Menu, select SQL Defaults.
3	Press Enter. Result: The SQL Defaults panel displays.
4	On the SQL Defaults panel, enter one of the following values to bypass display of the Statements panel: <ul style="list-style-type: none"> ● Type E to EXPLAIN all statements in the input file without gathering statistics. ● Type G to EXPLAIN all statements in the input file and gather statistics.
5	If you have the authority to permanently save your selection, type Y in the Permanent field. Result: !DB/EXPLAIN saves your entry and bypasses the Statements display for all subsequent sessions.
6	Press Enter. Result: Your current session goes directly to the EXPLAIN display. All statements are EXPLAINed. Statistics are or are not gathered depending on the value you entered on SQL Defaults.

Access from Candle's OMEGAMON II for DB2 (First Access)

Overview

This unit provides information about accessing !DB/EXPLAIN from OMEGAMON II for DB2.

(If you do not plan to access !DB/EXPLAIN from the primary menu for OMEGAMON II for DB2, you do not need to perform the procedures in this unit.)

Background about accessing !DB/EXPLAIN from OMEGAMON II for DB2

You can access !DB/EXPLAIN from Candle's OMEGAMON II for DB2. First, access Candle's !DB/Tools through OMEGAMON II's Primary Options Menu. Then select !DB/EXPLAIN from that menu.

Before the first such access, review the information contained in this unit. Once you complete the preparations in this unit, you can use OMEGAMON II with !DB/EXPLAIN by following the procedures in the unit "Using !DB/EXPLAIN with OMEGAMON II for DB2" on page 191.

Requirements for access

To access !DB/EXPLAIN from the primary option menu for OMEGAMON II for DB2, select option 3. When you select option 3, the KTC CLIST is invoked. You must be sure that the KTC CLIST is allocated to your SYSPROC concatenation.

If the KTC CLIST is not correctly allocated, OMEGAMON II for DB2 displays an informational panel that describes the tools instead of displaying the !DB/Tools Product Selection Menu.

Expanded information on preparing OMEGAMON II to access !DB/EXPLAIN can be found in the *!DB/Tools Installation and Customization Guide* in the chapter "Requirements for Access to !DB/Tools from OMEGAMON II for DB2."

Subsequent Accesses to !DB/EXPLAIN from OMEGAMON II

For expanded information about the interactions of OMEGAMON II for DB2 and !DB/EXPLAIN, see "Using !DB/EXPLAIN with OMEGAMON II for DB2" on page 191. Topics include toggling between OMEGAMON II in CUA mode and !DB/EXPLAIN running under TSO, and accessing OMEGAMON II from the !DB/EXPLAIN Plans panel.

Accessing !DB/EXPLAIN Batch Facilities

Overview

This unit provides information about accessing !DB/EXPLAIN in batch. It also describes where you can find additional information on running !DB/EXPLAIN in batch mode.

Background about using batch

When performing a large amount of work, Candle recommends using !DB/EXPLAIN's batch facility to obtain information or perform tasks in order to eliminate contention and improve system performance. JCL to run !DB/EXPLAIN batch jobs can be created by using the JCL command. See “Using the JCL? Command to Create a Batch Job” on page 407 for information. !DB/EXPLAIN provides jobs to:

- Extract data from the DB2 catalog or PLAN_TABLE
 - Manage plans, packages, and DBRMs
 - EXPLAIN plans, packages, statements, and DBRMs
 - EXPLAIN SQL statements, including those associated with BIND commands, and DBRMs, prior to their being bound
-

Sources of information

The JCL to run !DB/EXPLAIN batch jobs differs depending on the task you want to perform.

Expanded information on !DB/EXPLAIN's batch jobs can be found in these sections:

- “The !DB/EXPLAIN Extract Program and the Extract ID” on page 333
- “Using !DB/EXPLAIN in Batch” on page 401
- “Batch Processing for Application Development” on page 419
- “Batch Processing for Database Administration” on page 439

Exiting !DB/EXPLAIN

Background about exiting !DB/EXPLAIN

You can use the !DB/EXPLAIN configuration options (available in Housekeeping) to control the operation of the RETURN command and define it in your profile. You can define the RETURN command to have it function in two ways:

- Display the !DB/EXPLAIN Primary Menu
- Exit !DB/EXPLAIN

Exiting the product

The operation of RETURN as described in this procedure assumes that your user configuration has RETURN defined to exit !DB/EXPLAIN.

Follow this step to end a session and exit !DB/EXPLAIN.

Step	Action
1	<p>Press Return.</p> <p>Result: The result varies based on your previous action:</p> <ul style="list-style-type: none"> ● If you accessed !DB/EXPLAIN from OMEGAMON II for DB2, !DB/EXPLAIN ends all other sessions and returns to the primary options menu for OMEGAMON II. ● If you accessed !DB/EXPLAIN from the !DB/Tools Product Selection Menu, !DB/EXPLAIN ends all sessions and returns to the !DB/Tools Product Selection Menu. ● If you accessed !DB/EXPLAIN directly, the system exits !DB/EXPLAIN.

Chapter 5.

Accessing !DB/EXPLAIN Functions

Introduction

This chapter introduces you to the !DB/EXPLAIN Primary Menu from which you can select !DB/EXPLAIN objects against which to perform !DB/EXPLAIN functions, and the Administration Menu, from which you can perform various administrative tasks. The chapter tells you how to initiate multiple !DB/EXPLAIN sessions, and directs you to sources of information about the three basic menus you use to manipulate !DB/EXPLAIN objects: commands, selects, and filters.

Chapter contents

Access from the Primary Menu	108
Using the Administration Menu	110
Using the JCL Generation Menu	112
Access from the Sessions Menu	114
Methods for Working with Sessions	116
Restrictions Applicable to Working with Sessions	118
Using the Functions of the !DB/EXPLAIN Online Menu	119
Processing Multiple Datasets on the Online Menu	122
Processing Multiple Members on the Online Menu	123
Activities You Can Perform from a Typical !DB/EXPLAIN Panel	125

Access from the Primary Menu

Overview

This unit describes the selections available from the Primary Menu.

Accessing the Primary Menu

You access the Primary Menu by entering !DB/EXPLAIN through the !DB/Tools Primary Menu, through the !DB/EXPLAIN CLIST KTE, or through OMEGAMON II for DB2. It is the first !DB/EXPLAIN panel you see.

Elements of the Primary Menu

!DB/EXPLAIN's Primary Menu is illustrated below. Access the !DB/EXPLAIN function you want by typing its number in the Option field and pressing Enter.

```

----- DB/EXPLAIN 1 DB2=D42B 2 -----
Cmd ==> 3

                P R I M A R Y   M E N U 4

Option ==> 5

    1 Plans
    2 DBRMs
    3 Packages
    4 Collections
    5 Tables
    6 Libraries
    7 Costs
    8 Exceptions

    9 JCL Generation 7
   10 Dynamic SQL 8
   11 Administration Menu 9

    10 Extract ID      ==> D42B___
    11 Display Masks? ==> Y
    12 Masking in effect? ==> N

                                     13 Version: 500  Maint. Level: 9602  PSP: 2
----- (C) Copyright CANDLE Corp. 1993 - 1999 -----

```

Elements of the Primary Menu (continued)

The information below is keyed to the Primary Menu panel.

- 1** The product title centered and in upper case.
- 2** ID of the current DB2 subsystem. In the example, “DB2=” on the first line indicates the DB2 subsystem that is used in response to all “active” DB2 requests such as BIND, REBIND, and EXPLAIN.
- 3** Command line; the location for the entry of a !DB/EXPLAIN, ISPF, or TSO command.
- 4** The panel title centered and in upper case.
- 5** Area to enter the number of the !DB/EXPLAIN function you want to access.
- 6** The object list panels you can access from the Primary Menu.
- 7** The selection you make to display !DB/EXPLAIN's JCL Generation panel (described in “Using the JCL Generation Menu” on page 112).
- 8** The selection you make to display a panel that permits you to enter an SQL statement and execute it against DB2.
- 9** The selection you make to review and make selections from !DB/EXPLAIN's Administration Menu (described in the next unit).
- 10** The extract ID !DB/EXPLAIN is currently running. Typing ? in this field results in a display of allowable extract IDs.
- 11** Field in which to specify whether !DB/EXPLAIN displays the Selection Masking panel.
- 12** Field that indicates whether selection masking is in effect.
- 13** The maintenance level of the current system.

Using the Administration Menu

Overview

The functions on the Administration Menu were formerly available from the !DB/EXPLAIN Primary Menu. They have been separated so as to provide an additional administrative option and to improve readability. This unit describes the selections available from the Administration Menu.

Accessing the Administration Menu

You access the Administration Menu from the !DB/EXPLAIN Primary Menu option or by typing the command **ADMIN** on the command line of a !DB/EXPLAIN object list panel.

Elements of the Administration Menu

!DB/EXPLAIN's Administration Menu is illustrated below. To access the administrative function you want, type its number in the Options field and press Enter.

```
----- DB/EXPLAIN DB2=D42B -----  
Cmd ==>  
  
                A D M I N I S T R A T I O N   M E N U  
Cmds: GLOBAL (Menu) 1  
-----  
Option ==>  
  
    1 Activity Log 2  
    2 Authorizations 3  
    3 Extract Ids 4  
    4 Extract History 5  
    5 Housekeeping 6  
    6 Refresh displays from extract 7  
    7 Display DB/Tools profile data sets 8  
    8 Set DB/Tools profile data sets 9
```

Elements of the Administration Menu (continued)

- 1** The command you can enter from this menu to obtain the detailed global command menu
- 2** Displays the log of your activity; you can access the activity log from other !DB/EXPLAIN panels by entering the command **ACTLOG** on the command line.
- 3** If you are authorized to see them, displays a list of authorized users of !DB/EXPLAIN. Without appropriate authority, you cannot view or alter the list.
- 4** Displays the !DB/Tools Global Information—DB2 Subsystem Name Table panel that contains the extract IDs of configured DB2 subsystems available to you mapped against their associated subsystem IDs and MVS IDs
- 5** Displays extract history
- 6** Transfers you to the Housekeeping Main Menu
- 7** Rebuilds the displays generated from the extract
- 8** Displays information about profile datasets
- 9** Allows you to specify values for user and group profile datasets

Using the JCL Generation Menu

Overview

There are several methods to generate the JCL to run !DB/EXPLAIN functions in batch. This unit describes the JCL Generation Menu, one of the methods you can use.

Accessing the JCL Generation Menu

You access the JCL Generation Menu from the !DB/EXPLAIN Primary Menu option or by typing the command **JCL?** on the command line of a !DB/EXPLAIN object list panel. You can find more information about the JCL command in “Using the JCL? Command to Create a Batch Job” on page 407 in the chapter “Using !DB/EXPLAIN in Batch.”

Elements of the JCL Generation Menu

!DB/EXPLAIN's JCL Generation Menu is illustrated below. To identify the JCL you want to generate, type the appropriate number in the Options field and press Enter.

```

----- DB/EXPLAIN  DB2=DB2  -----
Cmd ==>

                J C L   G E N E R A T I O N

Option ==>

    1 DB/EXPLAIN Batch Utility          (KTEBUTIL)
    2 Explain SQL statements in a data set (KTEBSQLS)
    3 Explain BIND commands in a data set (KTEBDBRM)
    4 Explain SQL statements in a DBRM    (KTEBDBR2)
                                           ]
    5 Bind DB/EXPLAIN 2
    6 DB/EXPLAIN Extract 3
  
```

Elements of the JCL Generation Menu (continued)

- 1** Generates the JCL to run the batch program whose name appears in parentheses on the selected line
- 2** Generates the JCL to BIND plans and packages.
- 3** Generates the JCL to run an extract

Access from the Sessions Menu

Overview

This unit describes accessing !DB/EXPLAIN from the Sessions Menu.

Background about the sessions menu

The sessions menu is a “pop-up” menu that is accessed from the command line of any !DB/EXPLAIN panel. You can use the Sessions Menu as an alternative way of accessing !DB/EXPLAIN functions.

!DB/EXPLAIN permits you to have multiple sessions corresponding to !DB/EXPLAIN Primary Menu functions concurrently active and to switch between them. For example, you might have a session to work with plans and another to work with tables. When you switch from one session to another, the session you were using is suspended as you left it. The session you are entering becomes the active or “current” session.

Elements of the sessions menu

The sessions pop-up menu is illustrated below.

```
----- SESSIONS -----
Select ==> 1
2 1. // A Main Session
  2. /P Plans
  3. /D DBRMS
  4. /K C Packages
  5. /Q Collections
  6. /T Tables
  7. /L Libraries
  8. /C Cost
  9. /X Exceptions
----- 3 - 4 -----
```

Elements of the sessions menu (continued)

- 1** Location to enter the session number
- 2** Session number. The number you use to access a session from the Select line of the sessions menu
- 3** Session ID. The ID or alphabetic character (for example, P for plans or D for DBRMs) to use when accessing a session from the command line of any !DB/EXPLAIN panel
- 4** Status of the session:
 - C represents the current session—the session you are currently using.
 - A represents any session that is active but suspended.

Functions of the sessions menu

The sessions menu:

- Lists the types of sessions you can use
- Displays the status of the sessions
- Allows you to:
 - Begin a new session
 - Switch to an existing session

Background about the main session

The main session is the first session listed on the sessions menu. The main session is the session that begins when you access the sessions menu. The Main Session is “anchored” on the Primary Menu. It is denoted by two forward slashes (//) in the session ID column.

Effect of sessions on system resources

Using session switching is more costly in terms of available system resource usage than working from a single session.

Methods for Working with Sessions

Overview

This unit describes the methods you can use when working with multiple !DB/EXPLAIN sessions.

Method for accessing the Sessions Menu

If you want to use the Sessions Menu, follow these steps.

Step	Action
1	From the command line of any !DB/EXPLAIN panel, type a single forward slash (/).
2	Press Enter. Result: !DB/EXPLAIN displays the sessions menu.

Method to begin or switch a session

You can either begin a new session or switch to an existing session from the sessions menu or from any !DB/EXPLAIN panel command line. In either case, follow these steps.

Step	Action
1	To begin a session or switch sessions, either: <ul style="list-style-type: none"> ● Access the sessions menu. Type either the session number or the first letter of the object's name in the Select field. Example: If you want to start or switch to a DBRMs session, type 3 or D in the Select field. ● On any !DB/EXPLAIN panel command line, type <i>/id</i>. Example: If you want to start or switch to a DBRMs session, type <i>/D</i> on the command line.
2	Press Enter. Result: !DB/EXPLAIN suspends the session you were using and makes the selected session the current one. Example: !DB/EXPLAIN suspends the current session, displays the DBRMS panel, and makes the DBRMs session the current session.

Method for returning to your main session

To return to your main session, follow these steps.

Step	Action
1	On any object list panel, type // on the command line.
2	Press Enter. Result: !DB/EXPLAIN displays the main session. (!DB/EXPLAIN considers the session you left neither current nor active.)

Method for ending all sessions

To end all sessions, press END (usually PF3) as many times as necessary (or type END on the command line of each successive panel) until the !DB/EXPLAIN Primary Menu displays.

When you exit the Primary Menu, all sessions are terminated and you leave !DB/EXPLAIN. (!DB/EXPLAIN does not maintain multiple sessions after you exit the Primary Menu. You must reestablish additional sessions when you reenter the product.)

Restrictions Applicable to Working with Sessions

Overview

This unit describes the restrictions that are applicable to session switching, to the use of the REFRESH command with sessions, and to making changes to masks during a session.

Restrictions on session switching

Some functions of !DB/EXPLAIN do not permit the use of session switching. If you are using one of these functions and request to switch sessions, !DB/EXPLAIN displays a message indicating that session switching is not permitted. Some specific restrictions on session switching follow, but should not be considered all-inclusive.

Restriction on REFRESH command with open sessions

If you issue the REFRESH command and have open session windows, !DB/EXPLAIN issues a message telling you that the refresh cannot be performed. To perform a REFRESH when you have open sessions, follow the steps in the table below.

Step	Action
1	Close all open sessions.
2	Issue or reissue the REFRESH command.

Restriction on changing mask values with open sessions

If you change masking default using the Selection Masking panel, and you have open session windows, your selections will not take effect until all open sessions have been closed and a refresh performed as described in the previous section. !DB/EXPLAIN issues a message telling you that the mask defaults will not take effect until all the refresh is performed.

Using the Functions of the !DB/EXPLAIN Online Menu

Overview

In the previous chapter, “Accessing and Exiting !DB/EXPLAIN,” you learned how to access !DB/EXPLAIN using the Online Menu, KTEON. To review this information, see “Access Using the EXEC KTEON” on page 90. In this unit, you will find out more about the functions provided on the Online Menu.

Functions supported by the Online Menu

The Online Menu, KTEON, permits you to access !DB/EXPLAIN in either online or batch mode. From the Online Menu, you can invoke an EXEC or generate JCL to do the following:

- EXPLAIN SQL statements contained in one or more datasets
- EXPLAIN SQL statements associated with BIND commands in one or more datasets
- EXPLAIN SQL statements in DBRMs in one or more partitioned datasets

Selecting the object to EXPLAIN

This table summarizes the actions and results you obtain when working with the Online Menu panel to perform interactive EXPLAINS or generate JCL to perform batch EXPLAINS.

Select Option	To EXPLAIN	By Invoking EXEC...	or Generating JCL for Batch Job...
1	SQL statements contained in dataset <i>dataset_name</i> whose format is <i>format_code</i>	KTEXPLA	KTEBSQLS
2	SQL statements associated with BIND commands in dataset <i>dataset_name</i>	KTEXPLBA	KTEBDBRM
3	SQL statements in DBRMs that reside in dataset <i>dataset_name</i> . You also provide a DB2 <i>qualifier</i> for unqualified table names and, optionally, a <i>plan_name</i> .	KTEXPLD	KTEBDBR2

Background about specifying dataset names on the Online Menu

The dataset name you enter for functions 1 and 2 can be either a physical sequential or a partitioned dataset. When specifying option 3, DBRMS, the dataset you name must be partitioned.

Elements of the !DB/EXPLAIN Online Menu

This is the !DB/EXPLAIN Online Menu.

```

----- DB/EXPLAIN DB2=DB31 -----
Cmd ==>

                O N L I N E   M E N U

Extract ID      ==> 1          Estae Flag ==> Y 3
Create Batch JCL? ==> N ( Y Yes  N No ) 2

Specify Additional Data Sets? ==> N ( Y Yes  N No ) 4
Specify Member Lists?          ==> N ( Y Yes  N No ) 5

Option ==> 6
┌ 1 SQL Statements
└ 7 Data Set Name :
  File Format   :
┌ 2 BIND Commands
└ 8 Data Set Name :
  Allocate DBRMLIB?: ( Y Yes  N No  E Edit DBRMLIB concatenation)
┌ 3 DBRM
└ 9 Data Set Name :
  Owner/Qualifier :                      Plan:

----- (C) Copyright CANDLE Corp. 1993 - 1999 -----

```

- 1** Enter the extract ID you want to work with.
- 2** To use the Online Menu to invoke EXECs to perform interactive EXPLAINS, specify **N** in this field; to create JCL to perform the EXPLAINS in batch, specify **Y**.
- 3** Always set this field to **Y** unless directed to do otherwise by Candle.
- 4** To process only a single physical sequential or partitioned dataset during the current EXPLAIN execution or batch job, specify **N** in this field. To process multiple partitioned datasets, specify **Y**.

Elements of the !DB/EXPLAIN Online Menu (continued)

- 5** When processing partitioned datasets, you can process more than one PDS member at a time. Enter **Y** in this field to work with a list of members rather than only one or all members in the datasets you specify. Enter **N** in this field if you do not want to work with a list of members.
- 6** Enter the value **1**, **2**, or **3** in this field to EXPLAIN SQL statements, BIND commands, or DBRMs respectively.
- 7** Providing the required information in these fields and specifying **1** in the function field causes !DB/EXPLAIN to invoke EXEC KTEXPLA or generate the JCL to run batch job KTEBSQLS.
- 8** Providing the required information in these fields and specifying **2** in the function field causes !DB/EXPLAIN to invoke EXEC KTEXPLBA or generate the JCL to run batch job KTEBDBRM.
- 9** Providing the required information in these fields and specifying **3** in the function field causes !DB/EXPLAIN to invoke EXEC KTEXPLD or generate the JCL to run batch job KTEBDBR2.

Sources of additional information

This table identifies tasks for which you may want additional information and points you to units that address these tasks.

Task You Want to Perform	Source for Additional Information
Processing multiple datasets in a given !DB/EXPLAIN batch or online execution	“Processing Multiple Datasets on the Online Menu” in this chapter
Processing multiple members of one or more partitioned datasets in a given !DB/EXPLAIN batch or online execution	“Processing Multiple Members on the Online Menu” in this chapter
Using the JCL command to generate batch jobs based on information supplied on the !DB/EXPLAIN Online Menu	“Using the JCL? Command to Create a Batch Job” on page 407 in the chapter “Using !DB/EXPLAIN in Batch”

Processing Multiple Datasets on the Online Menu

Overview

This unit addresses how you can specify the use of more than one dataset during an online execution or batch job initiated from the !DB/EXPLAIN Online Menu.

Background about specifying multiple dataset names

If you plan to specify the use of multiple datasets during a given !DB/EXPLAIN execution initiated from the online menu, the datasets you specify can only be partitioned. They cannot be physical sequential.

How to specify the use of multiple datasets

If you want to perform an online EXPLAIN against more than one dataset, or if you want to create a batch job that operates against more than one dataset, follow these steps.

Step	Action
1	On the Online Menu, fill in these fields as follows: <ul style="list-style-type: none"> ● Type the number of the function you want to perform in the Function field. ● Type the required information in the fields for that function. Specify only one dataset name in the dataset Name field. ● Type Y in the Specify Additional datasets field.
2	Press Enter. Result: !DB/EXPLAIN displays the Specify <i>object_type</i> dataset panel, where <i>object_type</i> can be SQL, BIND Commands, or DBRMS.
3	On the Specify <i>object_type</i> dataset panel, fill in the names of any other partitioned dataset you want to process during this !DB/EXPLAIN execution.
4	Press Enter. Result: !DB/EXPLAIN initiates the online execution or creates the JCL required to run a batch job.

Processing Multiple Members on the Online Menu

Overview

This unit addresses how you can specify the use of a list of members during a !DB/EXPLAIN execution initiated from the Online Menu.

Requirements when specifying the use of a member list

To process a member list rather than a single member, the datasets you use must be partitioned. They cannot be physical sequential.

How to specify the use of a member list

Follow the steps in this chart to use member lists from the !DB/EXPLAIN Online Menu.

Step	Action
1	<p>On the Online Menu, fill in these fields as follows:</p> <ul style="list-style-type: none"> ● Type the number of the function you want to perform in the Function field. ● Type the required information in the fields for that function. Specify only one dataset name in the dataset Name field. ● Type Y in the Specify Member Lists? field. ● Optionally, to use multiple dataset, type Y in the Specify Additional dataset? field.
2	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Specify Member Lists panel.</p>
3	<p>On the Specify Member Lists panel, fill in the names of the members to be included or excluded from the current !DB/EXPLAIN execution or batch job. You can choose to use pattern masks to create your lists of members to be included or excluded.</p>
4	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN initiates the online execution or creates the JCL required to run a batch job.</p>

Background about member name patterns

Member name patterns can contain one or more pattern characters. These characters may be the percent sign (%) or the asterisk (*). The percent sign can represent only one character. The asterisk can represent any number of characters (including no characters). Thus, the pattern AB%Q can yield the member list ABRQ, AB3Q, or ABYQ. The pattern AB*Q can yield the member list ABQ, ABRQ, AB3Q, ABYQ, or ABZZZZQ.

Activities You Can Perform from a Typical !DB/EXPLAIN Panel

Overview

This unit identifies the basic activities you can perform and the elements of a typical !DB/EXPLAIN object list panel.

Performing basic activities

The basic activities you can perform from most !DB/EXPLAIN panels are listed in the table below. The table points you to the chapters of this guide where you can find additional information.

Activity	For information, see:
Execute a global command	“Using Commands” on page 127
Execute a command specific to a given object list or display (DO Command)	“Using Commands” on page 127
Execute a select	“Using Selects” on page 135
Sort the list of objects	“Using Sorts” on page 141
Filter the list of objects	“Using Filters” on page 147
Change the format of the panel	“Viewing Alternate Formats” on page 163
Customize !DB/EXPLAIN and set defaults	“Housekeeping” in the <i>!DB/EXPLAIN Reference</i>

Elements of a typical panel

A typical object list panel is illustrated below.

----- 1 DB/EXPLAIN 2 DB2=D31A ----- (F) 5 -----										
CMD ==> 4				SCROLL ==> CSR 6						
----- 3 D B R M S -----										
Cmds: DO (Menu) GLOBAL (Menu) 7				8 Selects: ? (Menu)						

	SEL	DBRM	VQCCMDH		STMTS	SELECTS	DELETES	UPDATES	INSERTS	LCK LCK
			LUOHXEL PLAN							XCL SHR

9	-	ALTSQLMU	NNNAN B	DBT925R	17	0	0	7	0	0 0 0
	-	ALTSQLUC	NNNAN B	DBT925R	14	0	0	1	2	0 0 0
	-	ALTSQLUM	NNNAN B	DBT925R	3	0	0	0	1	0 0 0

- 1** The product title centered and in upper case.
- 2** ID of the current DB2 subsystem. In the example, “DB2=” on the first line indicates the DB2 system that is used in response to all “active” DB2 requests such as BIND, REBIND, and EXPLAIN.
- 3** The panel title centered and in upper case
- 4** command line; the location for the entry of a !DB/EXPLAIN, ISPF, or TSO command
- 5** Message area. Short warning or error messages can appear in this area of the panel. On some panels, this area displays a line count for the current object list. An (F) in the message position indicates that filtering is in effect. An (M) in the message position indicates that masking is in effect.
- 6** The scrolling factor in effect
- 7** The types of command menus available from this panel. In this example, both global and local (DO) commands are available.
- 8** This field tells you to enter ? to see the Selects Menu. Specification of ? in the select field of a given object in the object list results in a display of the selects available on this panel.
- 9** Location for the entry of a given select against a specific object on the object list.

Chapter 6. Using Commands

Introduction

This chapter describes the types of commands available to you when you are in a !DB/EXPLAIN session and provides instructions for their use.

“Dictionary of !DB/EXPLAIN Commands” and “Commands Available from Object List Panels” in the *!DB/EXPLAIN Reference* provide summary information about !DB/EXPLAIN's commands. “Controlling the Destination of !DB/EXPLAIN Output” on page 319 discusses how you can control the destination for output produced as the result of issuing a command.

Chapter contents

Determining What Commands are Available	128
Viewing Available !DB/EXPLAIN Commands	129
Method for Using Commands	132
Commands for Working with the User PDS	134

Determining What Commands are Available

Overview

The unit contains information about the types of commands available to you from !DB/EXPLAIN panels and provides instruction for accessing and using the command menus, where applicable.

Procedure to determine what !DB/EXPLAIN commands are available

A typical !DB/EXPLAIN panel is shown here. Review the Cmds field (**1**) to determine whether global commands, local (DO) commands, or both, are available from a given panel.

```

----- DB/EXPLAIN DB2=D31A ----- LINE 10F 23(F)
CMD ==> BINDCNV                               SCROLL ==> CSR
                D B R M S
  Cmds: DO (Menu) GLOBAL (Menu) 1           Selects: ? (Menu)
-----
SEL DBRM      VQCCMDH          LCK LCK
              LUOHXEL PLAN      XCL SHR
-----
- DSDDDB2UP  NNNAN B DBT925E      9   1   0   0   1   0   1
- EXPCB02    NNNAN C EXPPLN10    25  0   0   1   0   0   0
- EXPCB04    NNNAN C EXPPLN4     25  0   0   1   0   0   0
    
```

Using TSO commands from a !DB/EXPLAIN panel

You can run CLIST and other TSO commands from the command line of most !DB/EXPLAIN panels. Review the information in the chart for the actions to perform.

Task You Want to Perform	Action
Run a CLIST or another TSO command	Type TSO <i>commandname</i> on the command line. Press Enter.

Viewing Available !DB/EXPLAIN Commands

Overview

This unit tells you how to view the command menu to determine what commands are available from a given panel.

Viewing global commands

Global commands are commands that are not specific to a given panel but can be used from most !DB/EXPLAIN panels. The Global Commands menu is scrollable and is tailored to the display from which you request it. Follow the steps in this table to view global commands.

Step	Action
1	On the command line of an object list panel, type GLO or GLOBAL to directly access the Global Commands menu.
2	Press Enter. Result: The system displays the scrollable Global Commands panel

Commands on the menu that do not have numbers

Several commands on the Global Commands menu have no numbers. These commands cannot be selected from the menu. However, they can be typed on the command line of an object list panel and executed directly.

- Typing **CANCEL** or **CAN** on the command line of an object list panel cancels any selects you entered.
- Typing **LOCATE** on the command line of an object list panel repositions an object list starting with the object for which the **LOCATE** was issued. **LOCATE** can be entered only on those panels where the **SORT?** command is also available.
- Typing **HEADERn** on the command line of an object list panel sets subheadings in a report of the current display.
- Typing **SQL** on the command line of an object list panel permits you to enter and execute SQL statements.

Global Commands menu

This figure shows the Global Commands panel.

```

----- DB/EXPLAIN DB2=DB31 -----
Cmd ==> GLO

          P R I M A R Y   M E N U
Option ==> ----- GLOBAL COMMANDS ----- ROW 1 TO 14 OF 46

Option ==> 1 JCL?  _ Access JCL generation menu      Scroll ==> CSR_
           2 JCL   Access JCL generation
           3 SET?  Access SET menu
              SETn Set SET option to #n
           4 SETC  Qualify unqualified tables with the
              Creator id specified on the SET?
              panel.
           5 SETO  Qualify unqualified tables with the
              qualifier the plan or package was bound
              with.
           6 SETU  Qualify unqualified tables with the
              TSO Userid.
----- PSP: 0 -----

```

Viewing DO commands

DO commands are commands that apply to the specific panel from which you accessed the DO Commands menu. An example DO Commands menu is shown in “Example DO Commands menu.” Follow the steps in this table to view DO commands.

Step	Action
1	<p>You can access the DO commands:</p> <ul style="list-style-type: none"> From an object panel. Type DO on the command line. From the Global Commands panel. Type the number of the DO command selection on the Global Commands panel command line.
2	<p>Press Enter.</p> <p>Result: The system displays the commands menu that allows you to select from all of the panel-specific commands you can use on a given panel.</p>

Example DO Commands menu

This figure is an example of a panel-specific commands menu. The menu is scrollable.

```

----- DB/EXPLAIN DB2=DB31 ----- LINE 1 OF 2,677
Cmd ==> DO                               Scroll ==> CSR
                                     D B R M S
Cmds: DO (Menu) GLOBAL (Menu)           Selects: ? (Menu)
----- DBRM COMMANDS -----
                                     ROW 1 TO 14 OF 70
SEL DBRM                                CK LCK
-----                                CL SHR
-----                                -- --
Option ==> _ Scroll ==> CSR_
- CICSDB2 1 INFO Show detailed DBRM information 0 0
- C335437 2 PLANS Display Plans 0 0
- C335437 3 DUPS Display All DBRMs 0 0
- C335437 4 NODUPS Reset DUPS command 0 0
- C335437 5 HIST Show DBRM History 0 0
- DB22DBR 6 NOHIST Reset DBRM History 0 0
- DB22DBR 7 COMPARE Compare latest EXPLAIN with results 0 0
- DB23DBR from a BIND EXPLAIN(YES) 0 0
- DB31DBR 8 IMPACT Impact Analysis 0 0
- DEDBAC0 9 COST Show Statement Costs 0 1
- DSDDB2U 10 XCPT Display Exceptions 0 1
- DSDDB2U 0 1
- DSDDB2U 0 1

```

Method for Using Commands

Overview

This unit tells you how to use commands from the command menus.

Method for using global or DO commands

To use a command from a Global or DO Commands menu, follow these steps.

Step	Action
1	On the menu, type the number for the command you want on the Command line.
2	Press Enter. Result: The system executes the process you requested.

Fast path for commands

Once you are familiar with !DB/EXPLAIN, you may know what command you want to execute without consulting either of the command menus. To use the fast path, follow the procedure below.

Step	Action
1	Type the name of the global or DO command you want to execute on the command line of the panel you are working with.
2	Press Enter. Result: The system executes the process you requested.

Additional information about commands

For information about the operation of specific commands listed on the menus, see the online help.

Contrasting commands and selects

When you use a global or DO command on a panel, it operates against all of the objects on that panel. (Note that an object list may be long enough to require several *physical* displays to include all of the objects on the logical panel.) This is in contrast to selects. A select performs many of the same functions as a command, but operates only on the object against which it has been entered. See “Using Selects” on page 135 for additional information on using selects.

Commands for Working with the User PDS

Overview

This unit describes two commands that enable you to edit or browse members in the user partitioned data set (PDS).

Background about the UEDIT and UBROWSE commands

!DB/EXPLAIN provides commands to permit you to view or edit members in your user PDS without exiting !DB/EXPLAIN or using a split screen. The commands are UBROWSE and UEDIT. To execute either of these commands, enter it on the command line of the object panel you are using.

Format of the commands

The commands UBROWSE and UEDIT take the form:

```
UBROWSE [member | member pattern]  
UEDIT [member | member pattern]
```

where

- *member* specifies the name of the member in the user PDS you want to browse or edit
- *member pattern* is a mask that is used to create a list of members which !DB/EXPLAIN presents to you. You can then select the member or members you want to browse or edit. Review the information in the section “Access Using the EXEC KTEON” on page 90 for information about creating pattern masks.

If you omit the member name or member pattern mask, !DB/EXPLAIN presents a list of the entire contents of the user PDS from which you can select the member or members to work with.

Chapter 7. Using Selects

Introduction

This chapter contains information about the selects you can use when working with !DB/EXPLAIN. Selects are characters that can be entered on a panel to perform similar functions to commands. However, instead of affecting all of the objects on a panel, a select affects only that object to which it is applied.

See “Dictionary of Selects” and “Selects Available from Object List Panels” in appendixes to the *!DB/EXPLAIN Reference* for summary information about !DB/EXPLAIN selects. “Controlling the Destination of !DB/EXPLAIN Output” on page 319 discusses how you can control the destination for output produced as the result of issuing a select.

Chapter contents

Information about Selects	136
Determining what Selects are Available	137
Fast Path for Using Selects on Object List Panels	138
Methods to Recover from Errors in Processing	140

Information about Selects

Overview

This unit describes how to determine what selects are available to you and how to use the Selects menu.

Contrasting commands and selects

When you use a global or DO command on an object panel, it operates against all of the objects on the panel. In contrast, a select performs many of the same functions as a command, but operates on only that object against which it has been entered.

A typical Selects menu

The example below illustrates a typical Selects menu.

```

----- DB/EXPLAIN DB2=DB31 ----- LINE 1 OF 2,677
Cmd ==>                               Scroll ==> CSR
                                D B R M S
Cmds: DO (Menu) GLOBAL (Menu)                Selects: ? (Menu)
----- DBRM SELECTS -----
                                ROW 1 TO 14 OF 34
SEL DBRM                               CK LCK
-----                               CL SHR
Option ==>                               Scroll ==> CSR_
- CICSDB2                               I Display detailed DBRM information           0 0
- C335437                               P Display all plans for a DBRM                 0 0
- C335437                               S List SQL statements for a DBRM               0 0
- C335437                               T Display Tables for a DBRM                   0 0
? C335437
- C335437                               C Compare latest EXPLAIN with results         0 0
- DB22DBR                               from a BIND EXPLAIN(YES)                     0 0
- DB22DBR                               J Compare DBRM attributes                     0 0
- DB23DBR                               0 0
- DB31DBR                               G Display cost details                         0 0
- DEDBAC0                               X Display Exceptions                          0 0
- DSDDDB2U                               W Perform Whatif analysis on a DBRM          0 1
- DSDDDB2U                               L Display latest EXPLAIN                     0 1
- DSDDDB2U                               H Display EXPLAIN history                     0 1
- DSDDDB2U                               0 1

```


Determining what Selects are Available

Overview

This unit describes the method to determine what selects you can use from a specific panel.

Displaying the Selects menu

Selects are panel-specific. To find out what selects are available on a given panel, use the procedure that follows to display the Selects menu.

Step	Action
1	In the SEL (select) column of a !DB/EXPLAIN panel, enter ? in the field next to one of the objects on the panel.
2	Press enter. Result: !DB/EXPLAIN displays the selections menu that allows you to choose from all the selects you can use on the panel.

Method for using the selects menu

If you want to choose a select from the menu, follow these steps.

Step	Action
1	On the Selects menu, type the appropriate value for the select you want in the Select field.
2	Press Enter. Result: The system applies the select to the object.

Additional information

For additional information about the operation of the selects listed on the menu, see the online help associated with the Selects menus.

Fast Path for Using Selects on Object List Panels

Overview

This unit includes information about using a select with one object or with multiple objects on a panel and how to display the results of multiple selections. It also describes how to cancel selections.

This unit is called the “fast path”, since you will enter selection codes directly on the object list panel.

Applying a select to one object

Once you are familiar with !DB/EXPLAIN, you may know what select you want to execute without consulting the select menu. To use the fast path, follow the procedure below.

Step	Action
1	Move the cursor to the line with the object you want.
2	Type the character for the selection you want in the Select field. Example: If you want to EXPLAIN a statement on the Statements panel, type E in the Select field.
3	Press Enter. Result: The system processes the select for the object and displays the results.

Applying selects to all objects on a panel

You can apply the same select to all objects on an object list panel using a command from the command line. Follow these instructions to apply a select to all objects on a panel. To use this facility, you must know the character that represents the select you want to use.

Step	Action
1	Move the cursor to the command line of the object list panel.
2	On the command line type S x , and press Enter, where <i>x</i> is the character representing the select you want to apply to all objects on the panel. The space between the S and the select character is mandatory. Example: On the DBRMS panel, type S L on the command line to display the latest EXPLAIN for all DBRMs on the panel.

Displaying the result of applying a select to all objects on a panel

In the previous example, you reequested a display of the latest EXPLAIN for all DBRMs on an object list panel by typing **S x** on the command line. To display the output of this example, follow these steps.

Step	Action
1	!DB/EXPLAIN displays the output for the first select. In our example, this is the latest EXPLAIN for the first DBRM on the object list panel.
2	Type END on the command line and press Enter, or press the appropriate function key. Result: !DB/EXPLAIN displays the results of processing the second select—the latest EXPLAIN for the second DBRM on the object list panel.
3	Continue to type END on the command line and press Enter, or press the appropriate function key until all of the DBRMs have been processed.
4	When the latest EXPLAIN for the last DBRM has been displayed, !DB/EXPLAIN redisplay the original object list panel.

If you do not want to complete the selections

If you have typed selects in the Select fields on the panel and decide that you do not want to proceed, follow these instructions.

Step	Action
1	Type CAN on the command line.
2	Press Enter. Result: The system blanks out all the select characters you typed and redisplay the panel.

Methods to Recover from Errors in Processing

Overview

This unit contains information about how to display more information about an error and methods available to recover from an error in processing of selects.

Displaying more information about an error message

If there is an error in the processing of any entry, the system displays an error message in the upper right corner of the panel. The system also positions the cursor on the select that the system cannot process. Perform this step to display more information about the error.

Step	Action
1	Type HELP on the command line and press Enter, or press the appropriate function key. Result: !DB/EXPLAIN displays expanded information about the error condition.
2	Some messages have another level of Help. When the long message is displayed, you can press the appropriate function key again. If additional help is available, it will be displayed.

Determining the appropriate option to use to resolve the error

Review the table to determine the appropriate option.

IF you want to ...	THEN ...
Use another select for the object	Type a different select code. Press Enter.
No longer use the select	Blank out the select. Press Enter.
Process the next select	Press Enter. Result: The system continues processing with the next select.

Chapter 8. Using Sorts

Introduction

This chapter contains information about the way to access the sorts you can use from an object panel.

See “!DB/EXPLAIN Sorts and Filters” in an appendix of the *!DB/EXPLAIN Reference* for charts that map the sorts available with !DB/EXPLAIN to the object list panels on which they can be entered.

Chapter contents

Defining the Default Sort	142
Accessing the Sorts Menu	143
Invoking a Sort	144
Returning a !DB/EXPLAIN Panel to Its Original Order ..	146

Defining the Default Sort

Overview

This unit provides information on how you can select a default sort sequence for applicable object list panels.

Procedure for defining default sorts

To define the default sort sequence for applicable object list panels, follow the steps in this table.

Step	Action
1	From !DB/EXPLAIN's Housekeeping Menu, select Panel Sorts.
2	Press Enter. Result: !DB/EXPLAIN displays the Panel Sorts panel.
3	Define the default sort sequence for the listed objects. For each object, in the Sort Type field: <ul style="list-style-type: none"> ● Type A to select an ascending sort sequence, or ● Type D to select a descending sort sequence.
4	For each object, in the Sort Command field, type the sort keyword you want to use as the default.
5	If you are appropriately authorized, type Y in the Permanent field to make the sort sequence permanent.
6	Press Enter. Result: !DB/EXPLAIN sets the default sort sequence and keywords.

Accessing the Sorts Menu

Overview

This unit describes how to access and use the Sorts menu.

Access from an object list panel

If you want to display a menu of the sorts that you can use with objects on the panel, follow these steps.

Step	Action
1	You can access the Sorts menu from an object list panel by typing DOS or SORT? on the command line.
2	Press Enter. Result: !DB/EXPLAIN displays a menu that provides a brief explanation of each sort and allows you to choose from all the sorts that you can use on the panel.

Access from the Global Commands menu

If you are on the Global Commands menu for a given object and want to display a menu of the sorts you can use, follow these steps.

Step	Action
1	Type the number that represents either of the commands DOS or SORT? on the command line of the Global Commands menu.
2	Press Enter. Result: !DB/EXPLAIN displays a menu that provides a brief explanation of each sort and allows you to choose from all the sorts that you can use on the panel.

Invoking a Sort

Overview

This unit tells you how to invoke a sort.

Invoking a sort from the Sorts menu

If you want to select a sort from the Sorts menu, follow these steps:

Step	Action
1	Type the number that represents the sort you want in the Sort Key field of the Sorts menu.
2	If you want to change the sequence of the sort, type the appropriate value in the Order field: <ul style="list-style-type: none"> • Type A for ascending order. • Type D for descending order.
3	Press Enter. Result: !DB/EXPLAIN sorts the data according to the sort key you selected and redisplay the data on the panel.

Using SORT commands from the object panel command line

To use a SORT command from a panel's command line, follow these steps.

Step	Action
1	Type one of the following on the command line. (Both the command and the keyword are required and there must be a space between them.) <ul style="list-style-type: none"> • If you want to sort in ascending order, type ASORT <i>keyword</i>. • If you want to sort in descending order, type DSORT <i>keyword</i>. <p><i>keyword</i> is the required acronym that specifies the field on which the sort will be performed. If you don't know the acronym for the sort you want, use the Sorts menu to initiate your sort.</p>
2	Press Enter. Result: !DB/EXPLAIN sorts the data according to the command and keyword you used and redisplay the data.

Synonyms for the SORT commands

This table shows synonyms for the ASORT and DSORT commands.

SORT Command	Synonym
ASORT	SORT or SORTA The following commands are equivalent: ASORT name SORT name SORTA name
DSORT	SORTD The following commands are equivalent: DSORT name SORTD name

Returning a !DB/EXPLAIN Panel to Its Original Order

Overview

This unit tells you how to return a panel to its original order.

Procedure

Review the chart to determine what method to use to return the display to its original order after you have used a sort.

Data You Have Sorted	Method to Use
Any sorted !DB/EXPLAIN data	Type SORT <i>keyword</i> on the command line and press Enter, where <i>keyword</i> is the original sort key. Result: !DB/EXPLAIN returns the data to its original order.

Chapter 9. Using Filters

Introduction

This chapter defines filtering and tells you how to use filters from !DB/EXPLAIN's filtering input panels and from the command line of the object list panels. Filtering permits you to subset the data that you view and work with.

See “!DB/EXPLAIN Sorts and Filters” on page 423 for charts that map the filters available with !DB/EXPLAIN to the object list panels on which they can be entered.

Chapter contents

Introduction to Filtering	148
Defining Default Filter Types	150
Invoking Filtering	151
Subsetting the Objects to Display	155
Resetting Filters	158
Effects of Filtering	159
Setting Up a Filtering Macro	160
Using a Filtering Macro	161

Introduction to Filtering

Overview

This unit describes the purpose of filtering and contrasts filtering with masking, another means to limit viewed or manipulated data.

Purpose of filtering

Filtering permits you to display only those objects meeting criteria that you specify. Using filters can increase your productivity and reduce errors. By filtering a list of objects, the subsequent filtered display becomes more meaningful and any actions performed affect only those objects displayed after the filtering.

You may want to filter an object list:

- To display a subset of the list so you can review information more easily.
 - To identify exceptions.
 - To limit the number of objects upon which processing is performed.
-

Determining whether filtering is in effect

If filtering is in effect, !DB/EXPLAIN displays an F in the right-hand corner of the displayed panel next to the line count; for example, **1 of 147(F)**. If both filtering and masking are in effect, !DB/EXPLAIN displays FM in the right corner of the panel.

Filtering vs. masking

Like masking, filtering permits you to limit the number of objects on which processing is performed. Masking is different from filtering in that data that is “masked out” is not present in storage. Data that is “filtered out” is present in storage, but is not displayed to you. Generally, masking and filtering can be seen as progressive stages of a winnowing process.

Masking

Masking:

- Can occur first during the extract process. You can define masks to subset object lists as a step in creating the extract file. Data masked during the extract does not appear in the extract file.
- Can also be performed against the extract file when objects are read into memory to create !DB/EXPLAIN's internal object lists. Again, the data masked out does not appear on the internal object lists.
- Improves performance and reduces overhead.
- If masks are changed on the Selection Masking panel while in !DB/EXPLAIN, you must perform a REFRESH to put the new masks into effect.

Filtering

Filtering:

- Is performed against the internal object lists that may already have been winnowed by masking during extract and during object list creation. In this case, the data is contained in the object list, but is not displayed when !DB/EXPLAIN presents the object list to you.
- Is fast. Filtering can be changed via a command or panel display input.

Source of additional information

For more information on masking see “Extract Masking” on page 348 in this guide, and “Selection Masking” in the *!DB/EXPLAIN Reference*.

Defining Default Filter Types

Overview

This unit provides information on how you can select the default filter type for applicable object list panels.

Procedure for defining a default filter type

To define a default filter type for applicable object list panels, follow the steps in this table.

Step	Action
1	From !DB/EXPLAIN's Housekeeping Menu, select Panel Filters.
2	Press Enter. Result: !DB/EXPLAIN displays the Panel Filters panel on which you can define the default filter type for the listed objects.
3	For each object, in the Filter Type field, type one of the following: <ul style="list-style-type: none"> ● A to select <i>and</i> filtering ● R to select <i>or</i> filtering, or ● 0 to turn filtering off If you are appropriately authorized, type Y in the Permanent field if you want to make the filtering options permanent.
4	Press Enter. Result: !DB/EXPLAIN sets the default filter types.

Invoking Filtering

Overview

This unit tells you how to set filtering criteria.

Background about setting filtering criteria

You can set filtering criteria by doing any of the following:

- Entering filtering criteria directly on the command line of an object list panel
- Using the filtering input panel for a given object list

Using Filters from the command line

If you know what filters are available for a given panel and know the associated keywords, you can type the keyword for the filters directly on the command line. The chart below shows the general procedure for using filters from the command line. (It does not cover the variety of wild cards and option codes available from !DB/EXPLAIN. These details are covered in “Subsetting the Objects to Display” on page 155)

Step	Action
1	On an object list panel, type the keyword for the filter, an option code, and a value on the command line and press Enter. Example: On the DBRMs panel, type D#PL>1 and press Enter. This example filters for all DBRMs having more than one plan.
2	Type FILTA (<i>and</i> filtering) or FILTR (<i>or</i> filtering) on the command line (see “Subsetting the Objects to Display” on page 155 below) to select <i>and</i> filtering or <i>or</i> filtering (or to override a default filter definition).
3	Press Enter to activate filtering. Result: !DB/EXPLAIN filters the object list according to the values you have set and displays the filtered object list.

Viewing the result of filtering from the command line

When you use filters from the command line, the results you see differ depending on the order in which you enter the filters and commands.

To see iteratively the results of filtering, enter:

```

Filtering command
Filter
Filter
Filter
    
```

To see only the end product of filtering, enter:

```

Filter
Filter
Filter
Filtering command (for example, FILT0 or FILTA)
    
```

Using the Filtering Input Panel

You can display the filtering input panel for a given object type by typing **FILT** or **FILT?** on the command line of an object list panel. General procedures for using filters from the filtering input panel follow:

Step	Action
1	Display the filtering input panel by: <ul style="list-style-type: none"> ● Typing FILT? or FILT on the command line of the object list panel. ● Selecting the option for FILT? on the Global Commands panel.

Using the filtering input panel (continued)

Step	Action
2	<p>Press Enter.</p> <p>Result: The filtering input panel for the object list is displayed.</p> <pre> ----- DB/EXPLAIN DB2-D31A ----- LINE1 OF 39 D B R M F I L T E R I N G CMD5: GLOBAL (Menu) ----- FILTER ==> 0 (A AND R OR O OFF) OP VALUE DBRM DBRM NAME DPLN PLAN NAME DLIB LIBRARY DVER VERS OP VALUE OP VALUE DMIXD MIXED (Y / N) DQUOT QUOTE (Y / N) DDC31 DEC31 (Y / -) DCHRS CHARSET (K / A) DCOMM COMMA (Y / N) DHLNG HOSTLANG (B/C/D/F/P/2) DMISS STMT MISSING? (Y / N) B ASM C COBOL D C DVRF VERIFIED? (Y/N/E/T) F FORTRAN P PL/I 2 COBOL2 OP VALUE DPCDT (PRECOMPILE TIME - YYYYMMDDHHMMSS) DEXDT (EXPLAIN TIME - YYYYMMDDHHMMSS) DMCST (MAXIMUM STATEMENT COST) DACST (AVERAGE STATEMENT COST) DTCST (TOTAL STATEMENT COST) D#PL (NUMBER OF PLANS) D#TL (TOTAL NUMBER OF STATEMENTS) ----- S Q L C O U N T S D B R M S T M T S ----- OP VALUE OP VALUE OP VALUE D#AL (ALTER) D#EC (EXECUTE) D#RL (ROLLBAC) D#CA (CALL) D#EX (EXPLAIN) D#SE (SELECT) D#CL (CLOSE) D#FT (FETCH) D#ST (SET) D#CM (COMMIT) D#GR (GRANT) D#SC (SET CON) D#CN (CONNECT) D#IN (INSERT) D#SD (SET DEG) D#CR (CREATE) D#LS (LOCK SHR) D#SH (SET HOS) D#DC (DCL CRS) D#LX (LOCK XCL) D#SP (SET PKG) D#DM (DCL STM) D#OP (OPEN) D#SR (SET RUL) D#DT (DCL TBL) D#PR (PREPARE) D#SS (SET SQL) D#DL (DELETE) D#RE (RELEASE) D#UP (UPDATE) D#DS (DESCRIB) D#RV (REVOKE) D#WH (WHENEVE) D#DR (DROP) D#TS (SEL+CSR) ***** BOTTOM OF LIST ***** </pre>

Using the Filtering Input Panel (continued)

Step	Action
3	<p>On the filtering panel, choose the object subset to be displayed by entering an option code and value next to the filters you want to use. (A list of option codes is found in “Option Codes” on page 156.)</p> <p>Example: On the filtering input panel, on the line following the filter D#PL, you can enter > in the Op field and 1 in the Value field. This example filters for all DBRMs with more than one plan.</p>
4	<p>If you have used more than one filter, you must determine whether you want <i>or</i> filtering or <i>and</i> filtering. An explanation of these options is found in “Subsetting the Objects to Display” on page 155. Set the field FILTER to the appropriate value.</p>
5	<p>Press Enter to save your selections.</p>
6	<p>Type END on the command line and press Enter, or press the appropriate function key.</p> <p>Result: !DB/EXPLAIN filters the object list according to the values you have set and displays the filtered object list.</p>

Subsetting the Objects to Display

Overview

This unit describes the commands, option codes, and wild cards that can be used to subset your object lists.

Filtering commands

Several commands permit you to further subset objects you want to display. The table below identifies the results you want and describes the commands used to achieve them.

Result You Desire	Action to Take
The filtered list must satisfy all of the filtering criteria you have specified.	Select the filtering keywords on the filtering panel. Then type the command FILTA on the command line and press Enter.
The filtered list must satisfy at least one of the filtering criteria you have specified.	Select the filtering keywords on the filtering panel. Then type the command FILTR on the command line and press Enter.
The filtered list must contain objects affected before or after a current timestamp.	<p>Select the filtering keywords on the filtering panel. Then type NOW (+/-nnnn) on the command line and press Enter.</p> <p>The example below filters for objects having a timestamp two days earlier than the current timestamp.</p> <p style="text-align: center;">NOW -2</p>

Option Codes

Filter criteria are specified by entering an option code (opcode) and a value.

Opcodes are:

=	equal to
≠	not equal to
>	greater than
≥	equal to or greater than
≤	less than or equal to
<	less than

Wild Cards

If you are filtering on a parameter that is of character type, you can use the following wild cards:

_ (underscore)	One-character wild card.
␣ (not sign)	The character is blank. Blanks must be specified as ␣.
* (asterisk)	One-character match if used in the middle of a string. Multiple-character match if used at the end of a string Note: The asterisk in QLOC * is not a wildcard.
% (percent)	One-character match if used in the middle of a string. Multiple character match if used at the end of a string.

Examples of the use of wild cards

Following are some examples of the use of wild cards.

- | | |
|--------------------|---|
| plan = * | Show all plans. |
| plan = dog% | Show all plans beginning with dog. |
| epref = - | Show all plans having an EXPLAIN where the prefetch column is blank |
| plan = d_g% | Show all plans with the first character d and the third character g . |
| plan = d-g% | Show all plans with the first character d , the second character blank, and the third character g . |

Resetting Filters

Overview

This unit describes the methods used to reset or turn off filtering.

Methods to use

To reset or turn off filters, take the actions described below:

Result You Desire	Action to Take
To reset a specific filter from the command line of an object panel or filtering input panel	Type the filter's keyword followed by the equal sign (=) and press Enter.
To reset a specific filter on the filtering input panel	Erase the value in the field and press Enter. Then either type END on the command line and press Enter, or press the appropriate function key.
To turn off all filters (temporarily)	<p>Type FILTO on the command line and press Enter. This turns off all filters but preserves their settings. These filters can be turned on again by entry of FILTA or FILTR on the command line.</p> <p>Thus, you can “toggle” between your filtered object list and your unfiltered object list by typing FILTO on the filtered list command line to see the unfiltered list and then typing FILTA or FILTR, whichever you previously specified, to return to the filtered list.</p>
To reset all filters	Type NOFILT on the command line and press Enter. This erases all of your previously specified filtering criteria.

Effects of Filtering

Overview

This unit describes the effects of specifying filtering when you are navigating from panel to panel, function to function, or between !DB/EXPLAIN sessions.

From panel to panel

If a given filter is in effect on multiple types of object lists, for example, DBRMs and Plans, then setting a filter on one such panel has the effect of applying that filter to other panels having that same filter.

From function to function

If you filter a Plans object list and then type **D** to obtain a list of DBRMs, *only* those Plans that were on the filtered display will have DBRMs displayed for them. Even though you may enter the command NOFILT on the DBRMs panel, you will not see DBRMs for Plans you previously filtered out.

Between sessions

Filtering you establish in one session of !DB/EXPLAIN is independent of filtering in another session you may be running simultaneously.

Setting Up a Filtering Macro

Overview

This unit provides instructions for setting up a filtering macro.

Background about setting up a filtering macro

If you have common filtering criteria (some things you always filter on), you can set up a macro to issue your filtering commands. Creating a macro makes it unnecessary to display the filtering input panel and type in all the filtering criteria every time you enter the product.

For general information on macros, see “Using the Macro Facility” on page 183.

An example filtering macro

You may want to see all the DBRMs which do DDL or DCL. The filtering commands to do this would be:

```

NOFILT      (Clear existing filters)
FILTR      (Set up OR filters)
D#CR > 0    (# of CREATEs > 0)
D#DR > 0    (# of DROPs > 0)
D#GR > 0    (# of GRANTs > 0)
D#RV > 0    (# of REVOKEs > 0)

```

Set-up procedure

To set up a filtering macro, follow the steps in the chart below.

Step	Action
1	Create a member in your User PDS containing the commands and filters you require. Refer to the example above as a model.
2	Name the member. Macro names must begin with a \$, for example, \$DBF0001.

Using a Filtering Macro

Overview

This unit describes the steps to use a filtering macro.

Procedure to use a filtering macro

To use a filtering macro, follow the steps in the chart below.

Step	Action
1	Type the macro's member name on the command line of the object panel you want to filter.
2	Press Enter. Result: The commands the macro contains are executed and your object list is filtered.

Chapter 10. Viewing Alternate Formats

Introduction

Some of !DB/EXPLAIN's object list panels contain so many fields you may want to display only a subset of them; or you may want to display them in an organization different from that of the primary display. This chapter discusses how to display !DB/EXPLAIN's panels in alternate formats using commands and menus.

Chapter contents

What is an Alternate Format?	164
Defining the Default Format	165
Accessing the Formats Menu	167
Viewing a Specific Format	169

What is an Alternate Format?

Overview

This unit tells you what an alternate format is.

Description of an alternate format

Some of the object list panels provide extensive amounts of information. For convenience or ease of understanding, or as the result of your information needs, you may want to see only a subset of the total information available for a given object. Alternatively, you may want to see the available information organized in a different way than the standard object list panel provides.

To satisfy this requirement, !DB/EXPLAIN provides predefined alternate formats for most object list panels.

Defining the Default Format

Overview

This unit provides information on how you can select the default format for applicable objects.

Procedure for defining a default format

The default format is that format which is displayed whenever you request display of an object list. For those !DB/EXPLAIN objects whose information can be displayed in multiple formats, you can define the format you want to use when displaying that object's information. To define a default format for applicable objects, follow the steps in this table.

Step	Action
1	From !DB/EXPLAIN's Housekeeping Menu, select the option for Panel Formats.
2	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Panel Formats panel on which you can define the default format number for the listed objects.</p> <div data-bbox="570 1085 1248 1551" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> P A N E L F O R M A T S PERMANENT? ==> (Y Yes N No) FORMAT # INITIAL CMD ----- COLUMN DISTRIBUTION 1 COLLECTIONS 1 COMPARE EXPLAIN 1 COSTS 1 DBRMS 1 ESTIMATOR 1 EXPLAIN 1 HISTORY 1 LIBRARIES 1 PACKAGES 1 PLAN 1 STATEMENTS 1 TABLE COLUMNS 1 VERIFICATION 1 WHATIF 1 ENTER to process END to cancel </pre> </div>

Procedure for defining a default format (continued)

Step	Action
3	For each object, in the Format # field, type the number of the format you want as the default. If you are appropriately authorized, type Y in the Permanent field if you want to make the Format # option permanent.
4	Press Enter. Result: !DB/EXPLAIN sets the default format number.

Accessing the Formats Menu

Overview

This unit describes how to access and use a Formats menu.

Example Format menu

One way to change the format of an object list display is to access and use the Format Menu for that display. An example Format Menu is shown here.

```

-----DB/EXPLAIN DB2=D31A -----
Option ==>   (Number)

  1. DBRM, STATUS, #PL, #STMT, #SEL, #DEL, #UPD, #INS, #LCK X, #LCK S
  2. DBRM, PRECOMPILE TIMESTAMP, LIBRARY, VERSION
  3. DBRM, #STMT, #SEL, #DEL, #UPD, #INS, #LCK X, #LCK S, COST
  4. DBRM, EXP TIMESTAMP, MAX COST, AVERAGE COST, TOTAL COST
  5. DBRM, #STMT, #SEL, #DCL CSR, #DEL, #UPD, #INS, #LCK X, #LCK S

ENTER to process  END to cancel

```

Access from an object list panel

If you want to display a menu of the formats that you can use with objects on the panel, follow these steps.

Step	Action
1	On the command line of an object list panel, type FORMAT? .
2	Press Enter. Result: !DB/EXPLAIN displays a menu of the formats available for the object list panel. The menu contains a brief description or list of the fields available on each alternate format.
3	On the Formats menu, enter the number of the format you want to see in the Cmd field.
4	Press Enter. Result: !DB/EXPLAIN displays a object panel formatted according to the selection you entered.

Access from the Global Commands menu

If you are on the Global Commands menu for a given object and want to display a menu of the formats you can use, follow these steps.

Step	Action
1	Type the number that represents the command <code>FORMAT?</code> on the command line of the Global Commands menu.
2	Press Enter. Result: <code>!DB/EXPLAIN</code> displays a menu of the formats available for the object list panel. The menu contains a brief description or list of the fields available on each alternate format.
3	On the Formats menu, enter the number of the format you want to see in the <code>Cmd</code> field.
4	Press Enter. Result: <code>!DB/EXPLAIN</code> displays a object panel formatted according to the selection you entered.

Viewing a Specific Format

Overview

This unit tells you how to view a specific format from either the Global Commands menu or the Formats Menu.

Viewing a specific format from the Global Commands menu

If you want to select a format from the Global Commands menu, follow these steps:

Step	Action
1	Access the Global Commands menu.
2	On the Global Commands Menu, in the Command field, type the number that represents the global command FORMATN.
3	Press Enter. Result: !DB/EXPLAIN displays the Formats menu for the object list panel. To select a specific format from the Formats menu, follow the instructions in the next section.

Viewing a specific format from the Formats Menu

If you want to select a format from the Formats Menu, follow these steps:

Step	Action
1	Access the Formats Menu as described earlier in this chapter.
2	In the Formats field of the Formats menu, type the number that represents the format you want.
3	Press Enter. Result: !DB/EXPLAIN displays the objects from the object list panel in the format you selected.

Using the FORMAT command from an object panel command line

To use a format command from a panel's command line, follow these steps.

Step	Action
1	On the command line of the object list panel, type FORMAT <i>n</i> , where <i>n</i> represents the format number of the format you want to see. If you don't know the number of the format you want, use the Formats menu to initiate your format.
2	Press Enter. Result: !DB/EXPLAIN formats the data according to the format number you used and redisplay the data.

Chapter 11. Using ISPF Facilities within !DB/EXPLAIN

Introduction

!DB/EXPLAIN uses ISPF facilities. This chapter provides a review of information about ISPF functions that are frequently used within !DB/EXPLAIN.

Chapter contents

Performing Some Common Operations under ISPF	172
Returning to the !DB/EXPLAIN Primary Menu	175
Controlling the Operation of Scrolling	176
Using the HELP Command to Display Information	179

Performing Some Common Operations under ISPF

Overview

This unit describes the function of many of the ISPF commands you use to control your session.

ISPF versions supported by !DB/EXPLAIN Version 500

The !DB/EXPLAIN Version 500 no longer supports ISPF Versions 3.1 and 3.2. All of the !DB/TOOLS Version 500 support ISPF Version 3.3 and above.

Using TSO commands from a !DB/EXPLAIN panel

You can run CLISTs and EXECs and execute TSO commands from any !DB/EXPLAIN panel. Review the chart for the actions to perform.

Task You Want to Perform	Action
Run a CLIST or EXEC, or execute a TSO command	On the command line, type TSO <i>commandname</i> , where <i>commandname</i> is the name of the CLIST or command you want to invoke.

Displaying the assignment of commands to function keys

The PFSHOW command allows you to display or suppress the display of PF keys on the bottom two lines of a panel. Displaying the PF keys can cover up the last two text lines on the panel. For example, if you use the PFSHOW command and then display the selects menu, you may not see the last two selects on the panel. Review the chart for the action to perform to display or suppress the display of PF keys.

Information You Want	Action
Assignment of function keys (those typically used in a standard ISPF environment)	Type the appropriate value on the command line: <ul style="list-style-type: none"> ● To display the key assignments, type PFSHOW. ● To suppress the key assignment display, type PFSHOW OFF. Press Enter. Result: !DB/EXPLAIN changes the display, including the display on the panel you are currently using.

Displaying the panel identifier

You can turn the display of the panel identifier on and off. You may need to know the panel identifier if you contact Candle Customer Support about the behavior of a specific panel. Review the information in this chart for the action to perform.

Information You Want	Action
Panel identifier	Type the appropriate value on the command line: <ul style="list-style-type: none"> ● To display the panel identifier, type PANELID. ● To suppress the panel identifier, type PANELID OFF. Press Enter. <p>Result: The system either displays or suppresses the display of the panel identifier on this panel and all subsequent panels in the session.</p>

Creating a second ISPF session

You can create a second ISPF session and run !DB/EXPLAIN on both sessions simultaneously if your TSO USERID region size has sufficient memory. Review the instructions in the chart for the actions to perform.

Task You Want to Perform	Action
Create another ISPF display session	Type SPLIT on the command line and press Enter, or press the appropriate function key.
Create another ISPF display session and another !DB/EXPLAIN session	Type SPLIT on the command line and press Enter, or press the appropriate function key. <p>Then start another !DB/EXPLAIN session (see “Access from the Sessions Menu” on page 114).</p>

Returning to the !DB/EXPLAIN Primary Menu

Overview

This unit shows how you can use the END and RETURN commands to return to the !DB/EXPLAIN Primary Menu or a previous panel. It also contains information about the different ways the RETURN command can function.

Variations in the operation of the RETURN command

You can use the !DB/EXPLAIN configuration option (available in Housekeeping) to define and control the operation of the RETURN command by means of your user profile. The RETURN command can function in one of two ways:

- It can display the !DB/EXPLAIN Primary Menu
 - It can exit !DB/EXPLAIN
-

Returning to the !DB/EXPLAIN Primary Menu or to a previous panel

Review the information in this chart for the action to end the current panel and return to the Primary Menu or to the previous panel. (The operation of RETURN as described in the chart assumes that your user configuration has RETURN defined to display the !DB/EXPLAIN Primary Menu.)

Task You Want to Perform	Action
End operations on the current panel and access the !DB/EXPLAIN Primary Menu	Type RETURN on the command line and press Enter, or press the appropriate function key. Result !DB/EXPLAIN processes selections made on the panel. It then processes the RETURN command to display the !DB/EXPLAIN Primary Menu.
End operations on the current panel and return to the previous panel that you were using	Type END on the command line and press Enter, or press the appropriate function key. Result !DB/EXPLAIN processes the END Command. It then processes all selections you have made on the panel and displays the panel you were using previously.

Controlling the Operation of Scrolling

Overview

!DB/EXPLAIN uses standard ISPF scrolling functions. If you are not familiar with ISPF scrolling, review the information in this unit to learn how to scroll backward and forward within a document or list, move to the beginning or end of a document or list, or alter the number of lines scrolled as the result of a scroll command.

Scrolling backward or forward

Review the instructions in the chart for the actions to scroll backward and forward.

Task You Want to Perform	Action
Display a section closer to the beginning of the document or list.	Type UP on the command line and press Enter, or press the function key for which the scroll up function is defined.
Display a section farther down from the beginning of the document or list.	Type DOWN on the command line and press Enter, or press the function key for which the scroll down function is defined.
Display a section <i>nnn</i> lines closer to the beginning of the document or list.	Type UP <i>nnn</i> on the command line and press Enter, where <i>nnn</i> represents the number of lines you wish to move toward the top of the document. Example: UP 20
Display a section <i>nnn</i> lines closer to the end of the document or list.	Type DOWN <i>nnn</i> on the command line and press Enter, where <i>nnn</i> represents the number of lines you wish to move toward the bottom of the document. Example: DOWN 5

Scrolling to the beginning of a document or list

For information about scrolling to the beginning of a document or list, review the chart for two of the possible actions.

Task You Want to Perform	Action
Display a section at the beginning of the document or list.	Type TOP (for Maximum Scroll Up) on the command line and press Enter, or press the appropriate function key.
	Type M on the command line and press the function key that you assigned to the UP command.

Scrolling to the end of a document or list

When the system displays the end of a document or list, it displays a full screen rather than just the last line. It also displays a message indicating that the display is the end of the text. For information about scrolling to end of a document or list, review the information in the chart for two of the possible actions.

Task You Want to Perform	Action
Display the section that is at the end of the document or list.	Type BOTTOM (for Maximum Scroll Down) on the command line and press Enter, or press the appropriate function key.
	Type M on the command line and press the function key that you have assigned to the DOWN command.

Altering the number of lines being scrolled

Review the information in the chart for the action to perform if you want to alter the number of lines that the program each time you request it to scroll.

Task You Want to Perform	Action
Alter the number of lines scrolled in response to a scroll command.	<p>Type the appropriate value in the scroll field.</p> <p>P (Page). The program scrolls a full page of information.</p> <p>H (Half). The program scrolls a half-page of information.</p> <p>C(Cursor). The program scrolls to the next cursor position.</p> <p><n>. The program scrolls the specified number of lines.</p> <p>Press Enter.</p> <p>Result: !DB/EXPLAIN scrolls up or down the number of lines you specified. The action you take affects scrolling on all panels including the one you are currently using.</p>

Using the HELP Command to Display Information

Overview

This unit contains information about accessing online help for !DB/EXPLAIN functions and messages, and for ISPF functions.

Displaying online help information

You can display help information for both !DB/EXPLAIN and ISPF facilities. Further information about !DB/EXPLAIN help can be found in the chapter “Resolving Problems” on page 457.

Review the information in this chart for the actions to display help information.

Information You Want	Action
Help for !DB/EXPLAIN functions	Type HELP on the command line and press Enter, or press the appropriate function key.
Help for ISPF facilities	On the panel displaying !DB/EXPLAIN help information, type HELP on the command line and press Enter, or press the appropriate function key.

Displaying online information about messages

This unit contains information about !DB/EXPLAIN messages that are displayed on a menu or panel.

Review this chart for the action to display information about messages.

Information You Want	Action
Information about a message that is displayed on the menu or panel	Type HELP on the command line and press Enter, or press the appropriate function key. Result: !DB/EXPLAIN displays more information about the condition.

Using the HELP Command to Display Information

Using !DB/EXPLAIN System Facilities

Chapter 12. Using the Macro Facility

Introduction

The macro facility helps increase productivity by saving commonly used !DB/EXPLAIN commands, selects, sorts, and filters as a single command that can be repeatedly reissued.

This chapter contains instructions for creating and using macros. It also describes the format and use of the example macros shipped with !DB/EXPLAIN. You can create your own macros and can use the examples as models.

Chapter contents

Background about Macros	184
Creating Macros	185
Invoking a Macro	186
Results of Executing a Macro	187
Example Macros Shipped with !DB/EXPLAIN	188

Background about Macros

Overview

This unit tells you what a macro is and how it can be used.

What is a Macro?

A macro is an executable member stored in your User PDS. It can contain any number of commands and selects combined together to perform a task.

Macro names

A macro name must begin with a dollar sign (\$) and is limited to eight characters in length.

Nesting macros

You can “nest” macros. This means that you can include a defined macro within the group of statements that comprise another macro. See “Example Macros Shipped with !DB/EXPLAIN” on page 188 for an example of nested macros shipped with !DB/EXPLAIN.

Creating Macros

Overview

This unit describes how to create a macro.

Prerequisites for creating a macro

When you create a macro, follow these instructions:

- Any macros you define as well as macros supplied as part of !DB/EXPLAIN must have names that begin with a “\$”.
 - Keep the macros you will use in your User PDS.
-

Tasks frequently performed when creating macros

Review the chart for the method to perform common tasks when creating macros.

Task You Want to Perform	Action When Creating the Macro
Add a comment to a macro	Type an asterisk (*) as the first character of each line of the comment. (You cannot continue a comment line using a plus sign.)
Apply the subsequent command to each selected object	Type a forward slash (/) between the functions. Example: S P/REBIND
Continue a line of a macro to the next line	Type a plus sign (+) as the last character of the line that is being continued.
Suspend processing of the macro for the current panel and wait for you to supply input	Type ? on a line by itself.

Invoking a Macro

Overview

This unit tells you how to invoke a macro during an online session or a batch run.

Prerequisites for invoking a macro

- Before using a macro to operate on an object, be sure that the commands, selects, sorts, and filters it contains are valid for the object type you have selected.
- Do not invoke a macro from the !DB/EXPLAIN primary menu.

Invoking a macro while online

To invoke a macro during an online session, type the macro name on the command line of the panel you are currently using and press Enter.

Invoking a macro in a batch run

Include the macro name within the group of statements you will use to act on a given object. The example below illustrates the use of a macro in a batch run.

Purpose	Example
Display all tables that the DBRM <i>dbrmname</i> accesses.	<pre> DB \$DISTAB (Object = DBRMs) (Macro containing the following statements: FILTA DBRM=<i>dbrmname</i> S T/PRNT/END (Terminate the example) END </pre>

You can find out more about running !DB/EXPLAIN in batch by referring to “Using !DB/EXPLAIN in Batch” on page 401.

Results of Executing a Macro

Overview

This unit describes what is output when a macro or macros are issued during online or batch processing.

Order of processing

When !DB/EXPLAIN encounters a macro, it processes the commands in the order they appear in the macro, but does not display any intermediate output.

Output during an online session

When a macro is processed during an online session, !DB/EXPLAIN displays the result of the last command in the macro.

Output during a batch run

When a macro is processed as part of a statement group during a batch run, the output of the macro can be input to subsequent commands in the statement group. Any output from the statement group is not printed or stored until the entire statement group has been processed.

Example Macros Shipped with !DB/EXPLAIN

Overview

This unit provides information about the example macros shipped with !DB/EXPLAIN.

Background about the example macros shipped with !DB/EXPLAIN

When you install !DB/EXPLAIN, example macros shipped in the CNTL dataset are copied into your User PDS. You can use them as they are and/or copy and modify them to meet your needs.

Example macros to be used online

These example macros are used during an online session of !DB/EXPLAIN

Name and Purpose	Invocation and example
<p>\$MACEX1: Set up filtering values for the DBRM panel.</p>	<p>Type \$MACEX1 on the command line of the DBRMS panel.</p> <p>FILTA (Indicates a FILTER) DBRM=D* (Only DBRMs starting with D) DTCST>0.0 (DBRM total cost > 0)</p>
<p>\$MACEX2: Display latest EXPLAIN for a selected DBRM. Set up formatting, perform a filter, sort resultant DBRMs by cost in ascending order, and then display the latest EXPLAIN for each DBRM.</p>	<p>Type \$MACEX2 on the Command line of the DBRMS panel.</p> <p>FORMAT3 FILTA (Indicates a filter) DBRM=D* (Only DBRMs starting with D) DTCST>0.0 (DBRM total cost > 0) ASORT DTCST (Sort ascending by total cost) S L (Select latest EXPLAIN)</p>
<p>\$MACEX3: Format the EXPLAIN panel.</p>	<p>Type \$MACEX3 on the command line of the EXPLAIN panel.</p> <p>EPATH (Access Path format) EPTBL (Access Plan Table format) ERECM (Access Recommendations format) FORMAT0</p>
<p>\$MACEX4: Filter DBRMs and format the EXPLAIN display using nested macros (a macro calling other macros).</p>	<p>Type \$MACEX4 on the command line of the DBRMS panel. When called by \$MACEX4, \$MACEX1 filters the DBRMs and displays the latest EXPLAINs using the format defined in macro \$MACEX3.</p> <p>\$MACEX1 (Call \$MACEX1) S L /\$MACEX3/ (Select latest EXPLAIN using \$MACEX3 format)</p>

Example macros to be used in batch

These example macros are used during a batch run of !DB/EXPLAIN.

Name and Purpose	Invocation and example
<p>\$MACEX5: Display the latest EXPLAIN for a selected DBRM. Set up formatting, perform some filtering, sort the resultant DBRMs by cost in ascending order, and display the latest EXPLAIN for the identified DBRMs using the format specified by the nested macro \$MACEX6.</p>	<p>Include the statement \$MACEX5 in the batch SYSIN deck following the primary command DB.</p> <pre> FORMAT3 FILTA (Indicates a filter) DBRM=ALTSQLUC (Only DBRMs named 'ALTSQLUC') DTCST>0.0 (DBRM total cost > 0) ASORT DTCST (Sort ascending by total cost) S L /\$MACEX6/ (Select latest EXPLAIN using \$MACEX6 format) END </pre>
<p>\$MACEX6: Set up formatting for the EXPLAIN panel.</p>	<p>This macro is nested within and invoked from \$MACEX5.</p> <pre> EPATH (Access Path format) EPTBL (Access Plan Table format) ERECM (Access Recommendations format) FORMAT0 PRNT END </pre>

Example Macros Shipped with !DB/EXPLAIN

Chapter 13.

Using !DB/EXPLAIN with OMEGAMON II for DB2

Introduction

By integrating !DB/EXPLAIN with Candle's OMEGAMON II for DB2, you have the most complete view possible for understanding and optimizing SQL and DB2 applications. The OMEGAMON II for DB2/!DB/EXPLAIN interface permits you to:

- Capture and analyze ad-hoc, dynamic SQL—SQL that is not stored in a place accessible to !DB/EXPLAIN users
- Analyze executing SQL
- Access OMEGAMON II application trace and historical reporting information to validate access path analysis

This chapter tells you how to make optimum use of the functionality these products offer when used together.

Chapter contents

Background about OMEGAMON II for DB2	192
Using OMEGAMON II and !DB/EXPLAIN Complementary Functions	193
Using the OMEGAMON II for DB2 Callable Interface . . .	195
Moving Interactively between !DB/EXPLAIN and OMEGAMON II	209
Using OMEGAMON II's Copy-and-Paste Facility	213
Accessing OMEGAMON II Trace Information from !DB/EXPLAIN	215

Background about OMEGAMON II for DB2

Overview

This unit provides an overview of OMEGAMON II for DB2 functionality.

Functions provided by OMEGAMON II

OMEGAMON II for DB2 provides a window into the operation of DB2. OMEGAMON II features:

- A CUA-compliant user interface for ease-of-use and faster, more intuitive problem resolution
- DB2 object and volume analysis that assists you to better understand and control DB2 I/O activity, identify problem DB2 objects, optimize dataset placement, and simplify SQL tuning
- A detailed view of thread activity and rule-based filtering to assist in identifying exception threads
- The Application Trace Facility (ATF) that looks at how an application accesses DB2 services during execution
- Complete historical information that includes reports and graphs of performance trace, accounting, statistics, audit, and DSNZPARM data
- Information about locking activity
- Information about DB2 connections to CICS, IMS, TSO, batch, and the Distributed Data Facility (DDF)
- Address space displays to assist in the analysis of factors associated with the DB2 address spaces
- Resource Manager analysis for the buffer manager, log manager, EDM pool, bind, SQL, lock, and DDF statistics

Using OMEGAMON II and !DB/EXPLAIN Complementary Functions

Overview

This section describes the functions of !DB/EXPLAIN and OMEGAMON II for DB2 that work together to help improve DB2 performance, and suggests ways you can use them.

Assumptions

The activities in this chapter presume the following:

- You have started an application trace for the application you want to review and the results of that trace are available. Trace results can be in memory or you can write them to data sets for longer term retention. Review your ATF information to ensure that:
 - It contains information about the plan or plans you want to study
 - It was run at a point in time that reflects the most current version of the plan or plans you want to study
- If you will use the historical reporting capabilities of OMEGAMON II for DB2 from !DB/EXPLAIN, you know the names of the VSAM data sets or DB2 tables that contain the OMEGAMON II stored accounting trace data. You can use the OMEGAMON II Historical Reporter OPTIONS command to specify the input data sets or DB2 table names to use for reporting, and the report filtering selection criteria.

When to use the complementary functions

Using !DB/EXPLAIN together with OMEGAMON II for DB2 assists you in resolving problems and making decisions related to an application's performance. Some example situations follow:

- You are running OMEGAMON II for DB2 and observe performance problems in an application. You want to analyze the SQL statements the application contains to determine what the problem may be.
- You are developing an application and want to:
 - Obtain predictive access path information from !DB/EXPLAIN about the SQL contained in the application
 - Obtain cost information from !DB/EXPLAIN about the SQL contained in the application
 - Verify the predictive information from !DB/EXPLAIN with the real-time execution information provided by OMEGAMON II's Application Trace Facility
- You are testing your application by executing an SQL call within !DB/EXPLAIN and want to review online performance information.
- !DB/EXPLAIN's review of an application highlights some potentially poorly-performing access paths and you want more detail about actual execution.

Using the OMEGAMON II for DB2 Callable Interface

Overview

The callable interface feature of OMEGAMON II for DB2 lets you make direct calls to !DB/EXPLAIN without having to first access TSO. This feature is not available with versions of OMEGAMON II for DB2 below 300.

Before using this feature, you should do the following:

1. Customize the Classic started task.
2. Set up !DB/EXPLAIN.
3. Set up the CUA options.
4. Set up the OMEGAMON II for DB2/!DB/EXPLAIN options.

Customizing the Classic Started Task to Implement the OMEGAMON II for DB2 Interface

Use the following procedure to customize the Classic started task.

Note: You may have already completed this procedure during the installation of OMEGAMON II for DB2.

Step	Action
1	<p data-bbox="570 1108 1390 1205">From the Configure OMEGAMON II for DB2 for Runtime Environment Default Menu, select option 2, Realtime Monitor Interfaces.</p> <div data-bbox="570 1220 1357 1612" style="border: 1px solid black; padding: 10px;"><pre data-bbox="597 1247 1336 1581">----- CONFIGURE OMEGAMON II FOR DB2 FOR RUNTIME ENVIRONMENT DEFAULT ----- OPTION ==> Select the following options to configure OMEGAMON II for DB2. You may type ALL to update the entire configuration or you may update specific sets of values by making selections from the list below. 1 DB2 Subsystem Information 2 Realtime Monitor Interfaces 3 Event Collection Manager and ATF configuration 4 Historical Components F1=Help F3=End</pre></div> <p data-bbox="570 1640 1330 1703">Result: The system displays the Started Task and VTAM Information Panel.</p>

Step	Action
2	<p>Press Enter.</p> <p>Result: The system displays the Classic Interface - DB/EXPLAIN panel.</p> <pre data-bbox="573 394 1356 888"> OMEGAMON II for DB2 ----- Classic Interface - DB/EXPLAIN ----- SSID: DIRT Option ==> The following information will be used to customize the DB/EXPLAIN interface. Specify: Configure? ==> Y (Y/N Y = Configure N= Do not configure) Started Task Name: TESTC ==> D2TE Maximum concurrent DB/EXPLAIN users: TEMAX ==> 10 Specify the following DB/TOOLS data set names: Messages ==> TDKT.DBTOOLS.MSGS Loadlib ==> TDKT.DBTOOLS.LOAD System Profile ==> TDKT.DBTOOLS.SYSTEM.PROFILE Group Profile ==> User Profile ==> TDKT.DBTOOLS.USER.PROFILE (Optional) (Optional) Press: ENTER to continue END for previous panel HELP for more information </pre>
3	<p>Specify the appropriate information on the Classic Interface - DB/EXPLAIN panel and press Enter.</p> <p>Result: The system returns you to the Configure OMEGAMON II for DB2 for Runtime Environment Default Menu.</p>
4	<p>Press the End function key.</p> <p>Result: The system displays the Configure OMEGAMON II for DB2 Menu.</p>
5.	<p>Use option 4, Create Runtime Members, to submit the job.</p> <p>This creates a started task in RKANPAR. The name of this started task is D2TE or the value entered on this panel for the parameter TESTC. Copy this started task to the JES2 procedure library for started tasks. This task dynamically allocates the !DB/EXPLAIN profile dataset. Configuration options for !DB/EXPLAIN will be obtained from this dataset.</p> <p>Note: Do not start this task. OMEGAMON II for DB2 starts this task whenever a !DB/EXPLAIN function is performed and stops it when the function has completed.</p>

Setting Up !DB/EXPLAIN

Use the following procedure to set up !DB/EXPLAIN for OMEGAMON II for DB2.

Step	Action
1	Ensure that !DB/EXPLAIN V500 PSP 9602 (or greater) is installed on every DB2 subsystem where you want to EXPLAIN statements. For information on installing !DB/EXPLAIN, see the <i>!DB/Tools for DB2 Installation and Customization Guide</i> .
2	<p>Define Destination Plan_Tables. The Destination PLAN_TABLE will be the current SQLID.PLAN_TABLE.</p> <ul style="list-style-type: none">● If the current SQLID.PLAN_TABLE does not exist, !DB/EXPLAIN attempts to create the tablespace PLANTBLS in the !DB/TOOLS default database.● If the PLAN_TABLE cannot be created there, !DB/EXPLAIN attempts to create the PLAN_TABLE in the database DSNDB04.● If it cannot create the table, !DB/EXPLAIN issues an error message indicating that the EXPLAIN could not be performed. <p>The current SQLID is determined by:</p> <ul style="list-style-type: none">● whether EXP_POWERX is defined in the !DB/Tools profile dataset concatenation as a non-blank value (for example, EXP_POWERX=SAMHILL), if so, the current SQLID is the value specified for EXP_POWERX,● or the current SQLID is determined by the DB2 security exit.

Setting Up the CUA Options

Use the following procedure to set up CUA options and to use the new !DB/EXPLAIN feature.

Step	Action
1	Log onto CUA.
2	<p>From the main CUA panel (KD2001), select Options from the task bar and press Enter. The following pop-up appears.</p> <div data-bbox="571 592 1356 850" style="border: 1px solid black; padding: 10px;"> <pre> 1. Preferences... 2. Controls... 3. Set thresholds... 4. User authorities... 5. User Profiles... 6. Subsystem Profiles... 7. Near-term Thread History Filter... 8. D2/Explain Options... F1=Help F12=Cancel </pre> </div>
3	<p>Select Option 2, Controls, from the pop-up and press Enter. Use the fields on the Terminal Options panel, shown below, to specify preferences for terminal options. For EXPLAIN, the key field on this panel is the SQL Process Level field. Type EXPLAIN or SAVE/EXPLAIN in this field to enable the function.</p> <div data-bbox="571 1108 1356 1480" style="border: 1px solid black; padding: 10px;"> <pre> Terminal Options Type a new setting or press Enter. Auto Refresh Interval . . . OFF Sec. (5-999 / Off) Use Status Words YES + (Yes/No) Critical Status Word Crit Warning Status Word Warn Normal Status Word OK Idle Status Word Idle SQL Process Level EXPLAIN <--- Invokes Explain SQL Process PF Key F21 + with PF21 CUA/TSO Toggle Key F24 + F1=Help F4=Prompt F12=Cancel </pre> </div>

Using the Terminal Options Window

The following table describes the fields on the Terminal Options pop-up window.

Fields for Terminal Options	Description
Auto Refresh Interval	Amount of time between updates to panels that display performance information. OMEGAMON II for DB2 collects performance data continually and displays the new data automatically at the frequency you enter in this field.
Use Status Words	On color terminals, color status bars represent the performance of status items. You can also specify a word or a series of characters to appear for each level of status. The word or characters appear in a row within the colored status bar. On non-color terminals, only the word or characters appear.
Critical Status Word	Word you want to appear when OMEGAMON II for DB2 displays a critical status (red). For example: <i>Crit.</i>
Warning Status Word	Word you want to appear when OMEGAMON II for DB2 displays a warning status (yellow). For example: <i>Warn.</i>
Normal Status Word	Word you want to appear when OMEGAMON II for DB2 displays a normal status (green). For example: <i>OK.</i>
Idle Status Word	Word you want to appear when OMEGAMON II for DB2 is not monitoring an object (turquoise). For example: <i>Idle.</i>

Fields for Terminal Options	Description
<p>SQL Process Level</p> <p><i>Values for the SQL Process Level are described in the following table.</i></p>	<p>Type of processing to use for SQL text when pressing the SQL Process PF Key.</p> <p>You can process SQL from the following screens:</p> <ul style="list-style-type: none"> ● SQL Call being Executed (Thread Detail). See “EXPLAINing SQL Code” on page 207. ● SQL Text (ATF). You can access the SQL Text screen through the application trace facility. Select the SQL TEXT pushbutton from the Application Trace SQL Detail screen. ● Thread History Dynamic SQL Detail
<p>SQL Process PF Key</p>	<p>If you have set up !DB/EXPLAIN for direct access, pressing this PF key will directly invoke !DB/EXPLAIN. Otherwise, it will invoke the SQL copy and paste facility to perform the EXPLAIN.</p>
<p>CUA/TSO Toggle Key</p>	<p>Use to toggle between CUA and TSO sessions.</p>

The following table provides a list of fields and valid values for completing the SQL Process Level field on the Terminal Options panel.

Values for SQL Process Level	Description
MANUAL	<p>After you have selected the SQL text to be EXPLAINED, you must use the PF key defined (SQL Process Key) to copy the SQL. Then switch to TSO/ISPF (CUA/TSO toggle), and paste the SQL, again using the PF key you defined.</p> <p>Note: To use MANUAL or AUTO, a TSO/ISPF edit session must be active in the background.</p>
AUTO	<p>After selecting SQL text, use the SQL Process key to copy and automatically insert the text into the member currently in use in the TSO/ISPF edit session. The placement of SQL text depends on the cursor position within the TSO/ISPF session.</p> <p>Note: To use MANUAL or AUTO, a TSO/ISPF edit session must be active in the background.</p>
SAVE	<p>After selecting the SQL text, use the SQL Process key to copy and automatically save the SQL text to the partitioned dataset member defined on the Save SQL Text Parameters panel. Text will either overwrite the existing contents of the member or be appended to it, depending on the type of save defined on the same panel. This option does not require a TSO/ISPF edit session.</p>
EXPLAIN	<p>This option invokes !DB/EXPLAIN, which produces a variety of !DB/EXPLAIN reports. Use the pushbuttons to view the various report sections. Complete the required options for the EXPLAIN option on the !DB/EXPLAIN panel.</p>

Values for SQL Process Level	Description
SAVE/EXPLAIN	Use this option to perform SAVE and EXPLAIN functions. Complete the required fields for the SAVE/EXPLAIN option.

Setting Up the D2/EXPLAIN Options

Use the following procedure to set up the D2/EXPLAIN options.

Step	Action
1	If you are not already logged onto CUA, then do so now.
2	<p>From the main CUA panel (KD2001), select Options from the task bar and press Enter. The following pop-up appears.</p> <div data-bbox="573 558 1356 814" style="border: 1px solid black; padding: 10px;"> <pre> 1. Preferences... 2. Controls... 3. Set thresholds... 4. User authorities... 5. User Profiles... 6. Subsystem Profiles... 7. Near-term Thread History Filter... 8. D2/Explain Options... F1=Help F12=Cancel </pre> </div>
3	<p>Select Option 8, D2/Explain Parameters, from this panel and press Enter. Use the D2/EXPLAIN Options panel, shown below, to specify parameters required to invoke !DB/EXPLAIN during your OMEGAMON session.</p> <div data-bbox="573 1005 1356 1339" style="border: 1px solid black; padding: 10px;"> <pre> D2/EXPLAIN Options Extract Id : DB41 User Profile DSN . . . : Group Profile DSN . . . : SQL Statement PDS. . . : Member Name. : Type of Save : REPLACE + Default Qualifier. . . : SYSIBM Qualifier Option . . . : PROMPT + Gather Statistics. . . : YES + Initial Report : EPATH+ F1=Help F4=Prompt F12=Cancel </pre> </div>

Using the D2/EXPLAIN Options Panel

The following table provides a list of fields and valid values for completing the D2/EXPLAIN Options panel.

Field	Description
Extract ID	1- to 8-character field that identifies an extract of plans and packages. It represents a logical grouping that typically identifies a DB2 subsystem and an MVS system. An Extract ID must be configured within !DB/EXPLAIN before it is used, but an actual extract is not required for the EXPLAIN.
User Profile DSN	Optional field names the dataset containing the !DB/EXPLAIN user profile. Contains overrides to the !DB/EXPLAIN options specified in the group profile and the system profile.
Group Profile DSN	Optional field names the dataset containing the !DB/EXPLAIN group profile. Contains overrides to the !DB/EXPLAIN options specified in the system profile.
SQL Statement PDS	Name of the dataset where the SQL statement is saved. Must be a PDS. The SQL Process Level is specified on the Terminal Options panel under the Options pulldown (fastpath O2). If SAVE, the text is saved but !DB/EXPLAIN is not invoked. If SAVE/EXPLAIN, !DB/EXPLAIN saves the text and invokes the EXPLAIN function.
Member Name	Name to use when saving the SQL text into the SQL statement PDS. If omitted, the system generates a name. When you define a member, the system prompts for whether the member should be replaced or appended.

Field	Description
Type of Save	<p>Type of save to be executed when saving SQL text to the member specified within the SQL statement PDS. Values are REPLACE and APPEND.</p> <p>REPLACE writes over an existing member or creates the member if it does not currently exist. System default.</p> <p>APPEND concatenates the new SQL text to the existing number.</p>
Default Qualifier	<p>Qualifier to use to qualify unqualified tables, views, and aliases in SQL statements.</p>
Qualifier Option	<p>Type of qualifier. Values are USE and PROMPT. USE indicates the current Default Qualifier for unqualified SQL text. PROMPT indicates you want to be prompted for a Qualifier whenever an SQL statement is used.</p>
Gather Statistics	<p>Type YES to have !DB/EXPLAIN gather statistics; type NO to bypass.</p>

Field	Description
Initial Report	<p>Type the name of the initial !DB/EXPLAIN report to be displayed: ESTMT, EPATH, EPTBL, EOBJ, or ERECM.</p> <p><i>ESTMT</i> is the actual SQL statement text being explained. If there is an SQL error, SQL error text also appears. Use the pushbuttons to select from the other !DB/EXPLAIN report sections that follow.</p> <p><i>EPATH</i> is a narrative describing the access path for the SQL statement.</p> <p><i>EPTBL</i> reports the plan table entries for the SQL statement in tabular form.</p> <p><i>EOBJ</i> reports the DB2 catalog statistics for objects referenced by the SQL statement. This section is available if you specified Yes for Gather Statistics in the D2/EXPLAIN Options panel under the Option pulldown (fastpath 08).</p> <p><i>ERECM</i> provides suggestions for optimizing the SQL statement.</p>

EXPLAINing SQL Code

Use the following procedure to process SQL from the Thread Detail panel.

Step	Action
1	Log onto CUA.
2	<p>From the main CUA panel (KD2001), select Active Threads and press Enter.</p> <p>Result: The system displays the All Threads Connected to DB2 panel.</p> <div data-bbox="570 640 1356 997" style="border: 1px solid black; padding: 5px;"> <pre> Actions GoTo View Options Tools Help +----- 08/14/99 8:07:07 KD20ATCD All Threads Connected to DB2 (Some) System D41 Lines 1 to 4 of 4 +-----+-----+-----+-----+-----+-----+-----+ Planname Elapsed CPU % Status Getpg Update Commit Jobname +-----+-----+-----+-----+-----+-----+-----+ DSNTEP41 31.4m 05.3% IN-SQL-CALL 5153K 61490 0 DLKEX212 K02300HP 1.3d 00.0% NOT IN-DB2 0 0 0 TD02S03 K02300HP 2.1d 00.1% NOT IN-DB2 0 0 0 TD02S44 KTWPLNR 1.0s 00.0% IN-SQL-CALL 0 0 0 RBRIN3 +-----+-----+-----+-----+-----+-----+-----+ F1=Help F2=Keys F3=Exit F5=Refresh F6=Console **=Bkwd F8=Fwd F10=Action Bar F15=System Status F24=CUA/TSO </pre> </div>
3	<p>Select the PLANNAME you want and press Enter.</p> <p>Result: The system displays the Thread Detail panel. If SQL calls are being executed by the thread, the message (IN-SQL-CALL) also appears. The Thread Detail panel shows detailed information about the activity an individual thread or a parallel task, including wait time information.</p> <div data-bbox="570 1270 1356 1822" style="border: 1px solid black; padding: 5px;"> <pre> GoTo Options Tools Help +----- 08/14/99 8:09:18 KD2VTHDD Thread Detail System D41A +-----+-----+-----+-----+-----+ Thread: Plan : KTWPLNR Attach: TSO Userid . . : RBRIN3 Connid : DB2CALL Region ASID . . : 225 Corrid : RBRIN3 Authid : RBRIN3 DB2 Status : IN-SQL-CALL MVS Status : USING CPU Elapsed Time : 00:00:05.932 CPU Utilization . . . : Total CPU Time : 00:00:03.586 Total Parallel Tasks . : Total Stored Proc CPU: 00:00:00.000 Current Parallel Tasks : +-----+-----+-----+-----+-----+ IN-DB2 Times Total Current +-----+-----+-----+-----+-----+ Elapsed Time 00:00:05.928 00:00:05 CPU Time 00:00:03.584 00:00:03 Stored Procedure CPU Time 00:00:00.000 N/A +-----+-----+-----+-----+-----+ (IN-DB2 Times) < Waits > < Current Wait Resource > F1=Help F2=Keys F3=Exit F5=Refresh F6=Console **=Bkwd ***=Fwd F10=Action Bar F15=System Status F24=CUA/TSO </pre> </div>

Step	Action
4	<p>Select Option 3, SQL Call Being Executed, from the GoTo pulldown menu.</p> <p>Result: The system displays the SQL Call Being Executed panel.</p> <div data-bbox="573 422 1357 1031" style="border: 1px solid black; padding: 10px;"> <pre> GoTo Options Tools Help ----- 08/14/99 8:47:27 KD20SQLD SQL Call Being Executed System D41A ----- Thread: Plan : DSNESPCS Attach: TSO Userid . . : RBRIN3 Connid : TSO Region ASID . . : Corrid : RBRIN3 Authid : RBRIN3 ----- Lines 1 to 15 of 15 ----- SQL call is active, call information is as follows : Thread Status = IN-SQL-CALL SQL Request Type = DYNAMIC Total SQL Reqs = 2 SQL Call Type = OPEN CURSOR SQL DBRM Name = DSNESM68 SQL Statement Number = 00193 Collection ID = DSNESPCS SELECT A.PLNAME, SECTNO, STMTNO, CARD FROM SYSIBM.SYSSTMT A, SYSIBM.SYSPLAN B, SYSIBM.SYSTABLES C WHERE A.PLNAME = B.NAME AND A.STMTNO > C.CARD ORDER BY SECTNO DESC, STMTNO ASC ----- </pre> </div>
5	Press the Process SQL key.

Moving Interactively between !DB/EXPLAIN and OMEGAMON II

Overview

This unit tells you how to move interactively between OMEGAMON II for DB2 and !DB/EXPLAIN using the copy-and-paste facility to review and EXPLAIN poorly performing SQL. It also describes how you can use the full functionality of !DB/EXPLAIN.

What functions can you use?

When using OMEGAMON II and !DB/EXPLAIN together, you can use the following OMEGAMON II for DB2 functions:

- Application Trace Facility (ATF) or thread detail displays to isolate poorly performing SQL statements
- Copy-and-paste facility to move poorly performing SQL identified in OMEGAMON II to !DB/EXPLAIN for detailed analysis and recommendations

The cut SQL text is pasted in a form that is correctly formatted so that it can be EXPLAINed without further editing.

- Application Trace Facility to view actual access paths chosen by the DB2 optimizer and compare these with the access path information predicted by !DB/EXPLAIN
- CUA/TSO toggle function key to access the full range of !DB/EXPLAIN functions
- Historical Reports to review accounting performance information about application plans and packages that are most frequently used, consume the greatest amount of resources, or have the biggest negative impact on performance

Modes of operation

When using !DB/EXPLAIN and OMEGAMON II for DB2 together, you can operate in the following modes:

1. You can move back and forth between OMEGAMON II and !DB/EXPLAIN in real time.
 - Access OMEGAMON II by means of its CUA interface.
 - Use OMEGAMON II's copy-and-paste facilities to isolate poorly performing SQL statements and paste them to an ISPF member. !DB/EXPLAIN can then analyze the SQL using the ISPF edit macro KTEXPL or a CLIST such as KTEON.
2. You can access the full range of !DB/EXPLAIN functions from within OMEGAMON II for DB2.
 - Use the CUA/TSO toggle to jump to a TSO session in which you have started or can start !DB/EXPLAIN.
 - Use !DB/EXPLAIN as you would normally.
3. You can access OMEGAMON II Historical Reporter from !DB/EXPLAIN to compare actual access path information with !DB/EXPLAIN's predicted access path information.
 - The direction of the access is only one-way: from !DB/EXPLAIN to OMEGAMON II for DB2
 - You access OMEGAMON II Historical Reporter from the !DB/EXPLAIN Plans panel using the A or P Select.
 - To use the A (Application Trace Facility) Select, you must have previously run an ATF trace for the plans of interest and placed the results into a data set.
 - To use the P (Plan Accounting) Select, you must have previously used OMEGAMON II for DB2's Historical Collector to collect accounting trace data for analysis.

Prerequisites for real time access

The Application Trace Facility (ATF) of OMEGAMON II for DB2 provides detailed runtime performance information at the thread, unit of work, program, and SQL statement level. To move interactively between !DB/EXPLAIN and OMEGAMON II, you must be using OMEGAMON II for DB2's CUA interface. You must also ensure that you have completed the prerequisite activities for access described in the chapter "Accessing and Exiting !DB/EXPLAIN" in the unit "Access from Candle's OMEGAMON II for DB2 (First Access)" on page 104.

Procedure to use the full function of !DB/EXPLAIN

If you are using OMEGAMON II for DB2, there are two procedures to access !DB/EXPLAIN and use its full functionality. The first way is discussed in "Accessing and Exiting !DB/EXPLAIN" on page 69 in the unit "Subsequent Accesses to !DB/EXPLAIN from OMEGAMON II for DB2". The second procedure involves use of the PF key assigned as the CUA/TSO toggle and is documented in this table.

The procedure in the table assumes that you have not previously logged onto TSO from within the current OMEGAMON II for DB2 session. If you have already logged onto a TSO application from within OMEGAMON II, use the CUA/TSO toggle to toggle between OMEGAMON II and !DB/EXPLAIN.

Procedure to use the full function of !DB/EXPLAIN (continued)

Step	Action
1	<p>From the OMEGAMON II for DB2 System Status panel, press the CUA/TSO program function key.</p> <p>Result: One of the following can occur depending on the way in which your OMEGAMON II for DB2 system has been installed:</p> <ul style="list-style-type: none"> ● You may be prompted for a TSO APPLID. Ensure you know the APPLID for logging onto your TSO system. ● The TSO APPLID can be displayed for your confirmation.
2	<p>Start !DB/EXPLAIN using a TSO CLIST. See “Accessing and Exiting !DB/EXPLAIN” on page 69 for information.</p>
3	<p>As you toggle between OMEGAMON II and !DB/EXPLAIN TSO sessions, you are presented in each instance with the last-used display (that is, the panel you were on when you pressed the toggle).</p>
4	<p>When you are ready to exit !DB/EXPLAIN running under TSO for the last time, you must fully exit the product using the END key. If you merely toggle to OMEGAMON II, your !DB/EXPLAIN session continues to run.</p>

What happens when you cancel the OMEGAMON II session

If you cancel your OMEGAMON II for DB2 session, the !DB/EXPLAIN session is also cancelled.

Using OMEGAMON II's Copy-and-Paste Facility

Overview

This unit describes how to use OMEGAMON II's copy-and-paste facility to capture static or dynamic SQL and move it to !DB/EXPLAIN for analysis. This feature is for use with OMEGAMON II for DB2 versions below 300. If you are using OMEGAMON II for DB2 Version 300 or higher, the callable interface feature is recommended.

Prerequisites to using the copy-and-paste facility

Within OMEGAMON II for DB2, be sure you know the location of the PF keys defined for the copy-and-paste and the CUA/TSO toggle functions. The default PF keys are PF21 and PF24 respectively. You can choose to assign these functions to other PF keys if you wish. If you are using OMEGAMON II V270 at the current maintenance level, a profile variable is provided that permits you to select either manual or automated copy-and-paste. If you are using this profile variable, you may need to modify the instructions in this unit based on the value you specified.

Capturing SQL from a running thread

You can use OMEGAMON II's copy-and-paste facility to capture running SQL from DB2 control blocks by following the steps in the table.

Step	Action
1	From OMEGAMON II for DB2's System Status panel, select the connection (for example, IMS, CICS, TSO) you want to view. Also select Thread Activity under Alerts. Result: The Thread Detail panel displays
2	On the Thread Detail panel, select G4 (SQL Text). Result: The SQL Text panel displays
3	On the SQL Text panel, press the copy-and-paste PF key to isolate and save the SQL text you want to EXPLAIN. Result: OMEGAMON II displays a pop-up indicating that the SQL text has been saved.
4	The CUA/TSO toggle permits you to toggle back and forth between your OMEGAMON II session and a TSO session. Press the CUA/TSO toggle PF key to switch to a TSO session.
5	Initiate an ISPF session and access the data set member into which you will paste the saved SQL text.

Capturing SQL from a running thread (continued)

Step	Action
6	Press the PF key you have assigned as copy-and-paste to paste the saved SQL to the ISPF member.
7	Invoke the ISPF edit macro KTEXPL to EXPLAIN the SQL text. For information on using KTEXPL, see "Access Using the Edit Macro KTEXPL" on page 78.
8	When you have completed your work with !DB/EXPLAIN, return to OMEGAMON II by pressing the CUA/TSO PF key.

Capturing SQL using the ATF

You can use OMEGAMON II's ATF to capture SQL statements and move them to !DB/EXPLAIN using the copy-and-paste facility. Follow the summary steps in the table.

Step	Action
1	From OMEGAMON II for DB2's Application Trace Facility, select the SQL TEXT pushbutton on the Application Trace SQL Detail panel. Result: The SQL Text panel displays.
2	On the SQL Text panel, press the copy-and-paste PF key to isolate and save the SQL text you want to EXPLAIN. Result: OMEGAMON II displays a pop-up indicating that the SQL text has been saved.
3	The CUA/TSO toggle permits you to toggle back and forth between your OMEGAMON II session and a TSO session. Press the CUA/TSO toggle PF key to switch to a TSO session.
4	Initiate an ISPF session and access the data set member into which you will paste the saved SQL text.
5	Press the PF key you have assigned as copy-and-paste to paste the saved SQL to the ISPF member.
6	Invoke the ISPF edit macro KTEXPL to EXPLAIN the SQL text. For information on using KTEXPL, see "Access Using the Edit Macro KTEXPL" on page 78.
7	When you have completed your work with !DB/EXPLAIN, return to OMEGAMON II by pressing the CUA/TSO PF key.

Accessing OMEGAMON II Trace Information from !DB/EXPLAIN

Overview

This unit describes how you can obtain OMEGAMON II Application Trace and Accounting Report information when you are working in !DB/EXPLAIN.

Direction of access

The direction of access when obtaining OMEGAMON II Application Trace and Accounting Report information is always one-way: from !DB/EXPLAIN to OMEGAMON II, with OMEGAMON II operating under ISPF.

Prerequisites for using the OMEGAMON II trace information

Ensure the Application Trace Facility has previously been used to trace your applications and that you have current application and accounting trace information available to you in data sets whose names are known to you. You must know the name of the data set containing the history information for the facility you want to use. You can use the OMEGAMON II OPTIONS menu to specify the input data sets or DB2 table names to use for reporting or to change their names, and to specify the report filtering selection criteria.

Procedure to obtain current trace information

If the trace information you have is not current or does not contain information about the plan or plans you want to study, follow the steps below.

Step	Action
1	Toggle to OMEGAMON II using the appropriate PF key.
2	Start the Application Trace Facility writing to a VSAM data set.
3	Access !DB/EXPLAIN and select the plan or plans you want to study.

For information about using OMEGAMON II for DB2's Application Trace Facility, see *OMEGAMON II for DB2 Reference*, D253-4359.

Procedure

To perform the tasks, take the actions described in the table.

Task You Want to Perform	Action to Take
Access the OMEGAMON II Application Trace Facility from !DB/EXPLAIN	On the !DB/EXPLAIN Plans panel, in the SEL (selects) column, type A in the SEL field of the plan for which you want to display application trace information. Result: OMEGAMON II displays the Application Trace Facility Menu panel.
Access the OMEGAMON II Accounting Reports Facility from !DB/EXPLAIN	On the !DB/EXPLAIN Plans panel, in the SEL (selects) column, type P in the SEL field of the plan for which you want to display accounting report summary information. Result: OMEGAMON II displays the Accounting Reports Facility Historical Reporter Options panel.

Sources of information

Refer to the table for sources of information about OMEGAMON II for DB2's Application Trace Facility and Accounting Reports.

Task you want to Perform	Source of Information
Access the OMEGAMON II Application Trace Facility from !DB/EXPLAIN	<i>OMEGAMON II for DB2 Reference Manual, D253-4359</i> <i>OMEGAMON II for DB2 Tuning Guide for the Classic Interface, D255-4360.</i> <i>OMEGAMON II for DB2 User's Guide, D254-4361</i>
Access the OMEGAMON II Accounting Reports Facility from !DB/EXPLAIN	<i>OMEGAMON II for DB2 Historical Reporting Guide, D299-4362.</i>

Chapter 14. Support for Distributed DB2

Introduction

!DB/EXPLAIN provides the commands and selects you can use to EXPLAIN objects on a remote DB2 subsystem and to gather statistics from that subsystem. This chapter describes how to execute !DB/EXPLAIN in a distributed environment.

Chapter contents

Prerequisites for Using !DB/EXPLAIN Distributed Support	218
Remote Functions Available with !DB/EXPLAIN	220
Commands and Selects to Perform Remote EXPLAINS	221
Performing Remote EXPLAINS to Support Application Development	224
Commands to Gather or Update Remote Catalog Statistics	226
Commands to Perform Remote BINDS/REBINDs for Packages	227
Specifying the Remote DB2 Subsystem	228
Working With a Remote EXPLAIN	231
Some Usage Scenarios	233
Performing Remote Functions in Batch	235

Prerequisites for Using !DB/EXPLAIN Distributed Support

Overview

This unit describes the prerequisites you must accomplish to execute !DB/EXPLAIN in a distributed environment.

Prerequisite activities in DB2

Before you can use !DB/EXPLAIN's distributed support, you must perform these activities:

- Define any remote DB2 subsystem you will access from !DB/EXPLAIN to DB2's Distributed Data Facility (DDF).
 - Define your local DB2 subsystem to the remote DB2's DDF.
 - Create the DB2 Communications Database (CDB) on each applicable subsystem. It must contain the appropriate pointers to the other subsystem(s) in the network.
 - Start the DB2 subsystem on which the EXPLAIN will be performed.
 - Bind the packages you will use at both the local and the remote DB2 subsystems.
-

Location of remote DB2 subsystems

Remote DB2 subsystems can be on the same MVS image as the local subsystem, or on another MVS.

Additional information about DB2 prerequisites

For additional information about defining a remote subsystem to DB2's DDF and creating and populating the DB2 CDB, see the IBM manual entitled *IBM DATABASE 2 Administration Guide Volume 1*.

Prerequisite activities in !DB/EXPLAIN

Before you can perform !DB/EXPLAIN functions against a remote DB2 subsystem, you must perform these activities:

- Install !DB/EXPLAIN on any subsystems where you will perform EXPLAINS.
 - Review the value you have defined on Housekeeping's SET Defaults panel in the Default Remote Location field to determine whether it meets your requirements. For more information about this field, see "Specifying the Remote DB2 Subsystem" on page 228.
-

Licensing requirements

You must have a !DB/EXPLAIN license on every central processing unit (CPU) where you will run EXPLAINS.

Remote Functions Available with !DB/EXPLAIN

Overview

This unit describes the remote functions you can initiate using !DB/EXPLAIN and how you initiate them.

Functions available on a remote DB2 subsystem

You can perform these functions at a remote DB2 subsystem using !DB/EXPLAIN:

- EXPLAIN and gather statistics
 - EXPLAIN and omit gathering statistics
 - Gather catalog statistics without an EXPLAIN
 - Update catalog statistics either temporarily or permanently
 - Perform remote package binds/rebinds
-

Initiating a remote EXPLAIN

You can initiate a remote EXPLAIN from these !DB/EXPLAIN panels:

- Plans
 - DBRMs
 - Packages
 - Statements
-

Gathering and updating catalog statistics

You can gather catalog statistics from a remote DB2 subsystem and can update the remote catalog from these panels:

- Estimator
 - What-if
-

Remote BIND/REBIND

You can execute a remote bind/rebind for packages from the Packages panel.

Commands and Selects to Perform Remote EXPLAINS

Overview

This unit describes the commands and selects you use to perform remote EXPLAINS.

Commands used to perform remote EXPLAINS

Use this table to determine the commands you can use to perform a remote EXPLAIN.

Command	Panels On Which Available	Function
EXPLAR	DBRMs Packages Plans Statements	Performs and displays a remote EXPLAIN for all objects regardless of bind time. Does not gather catalog statistics.
EXPLOR	DBRMs Packages Plans	Performs and displays a remote EXPLAIN for only those objects that have not been EXPLAINED since they were last bound. Does not gather catalog statistics.
EXPLR	DBRMs Packages Plans	Performs a remote EXPLAIN for only those objects that have not been EXPLAINED since they were last bound. Displays the latest EXPLAIN for all objects including those that have been previously EXPLAINED.
GEXPLAR	DBRMs Packages Plans Statements	Performs and displays a remote EXPLAIN for all objects regardless of bind time. Gathers catalog statistics.

Commands used to perform remote EXPLAINS (continued)

Command	Panels On Which Available	Function
GEXPLOR	DBRMs Packages Plans	Performs and displays a remote EXPLAIN for only those objects that have not been EXPLAINED since they were last bound. Gathers catalog statistics.
GEXPLR	DBRMs Packages Plans	Performs a remote EXPLAIN for only those objects that have not been EXPLAINED since they were last bound. Displays the latest EXPLAIN for all objects including those that have been previously EXPLAINED. Gathers catalog statistics.
WIFR	What-if	Temporarily updates the remote catalog. Performs an EXPLAIN using the new statistics with an option to make the statistics permanent.
WIFUR	What-if	Permanently updates the remote catalog and performs an EXPLAIN using the new statistics.

Selects used to perform remote EXPLAINS

Use this table to determine the selects you can use to perform a remote EXPLAIN.

Select	Panels On Which Available	Function
@	DBRMs Packages Plans Statements	Performs and displays a remote EXPLAIN for the object. Gathers catalog statistics.
#	DBRMs Packages Plans Statements	Performs and displays a remote EXPLAIN for the object. Does not gather catalog statistics.

Performing Remote EXPLAINS to Support Application Development

Overview

When you are developing an application, you may want to run an EXPLAIN against the SQL code in a module or data set. This unit describes how you can run an EXPLAIN against such SQL code contained in a nonextracted entity.

Definition of a nonextracted entity

A nonextracted entity is an entity for which an extract has not been run, for example, SQL statements in a data set or DBRM, or SQL statements to support BIND commands.

Procedure to perform a remote EXPLAIN for a nonextracted entity

To perform a remote EXPLAIN for a nonextracted entity, follow the steps in this table.

Step	Action
1	Access the member containing the SQL statements you want to EXPLAIN.
2	Execute the appropriate !DB/EXPLAIN edit macro or CLIST by typing its name and any required parameters on the command line of the member. (Instructions for executing these edit macros or CLISTs are in the chapter “Accessing and Exiting !DB/EXPLAIN” on page 69.)
3	Press Enter Result: !DB/EXPLAIN displays the Statements panel for statements contained in the member.
4	On the Statements panel, type the appropriate command or select to perform the remote EXPLAIN.

Commands and selects to perform a remote EXPLAIN for nonextracted entities

Use this table to determine which command or select to use when performing an EXPLAIN for a nonextracted entity.

Task You Want to Perform	Command to Use
Perform a remote EXPLAIN; do not gather statistics.	<ul style="list-style-type: none">● EXPLAR command● # select
Perform a remote EXPLAIN; gather statistics.	<ul style="list-style-type: none">● GEXPLAR command● @ select

Commands to Gather or Update Remote Catalog Statistics

Overview

This unit describes the commands you can use to gather statistics from the catalog of a remote DB2 subsystem or update the remote catalog's statistics.

Gathering and updating catalog statistics at a remote subsystem

If you want to perform an EXPLAIN in conjunction with gathering catalog statistics, see the previous unit. If you want to gather or update catalog statistics at a remote subsystem without performing an EXPLAIN, use a command in this table.

Command	Available From...	Function
GSTATS	Estimator What-if	Gather statistics from the remote catalog and refresh the display.
UPDTR	Estimator What-if	Update the remote catalog with new statistics.

Commands to Perform Remote BINDs/REBINDs for Packages

Overview

This unit discusses the process to bind/rebind packages on a remote subsystem.

Specifying the remote subsystem on which to bind a package

To specify the remote DB2 subsystem where a package is bound, type the subsystem ID of that subsystem in the Location field of the BIND/REBIND Package panel.

Overriding specification of the remote subsystem

You can override the location on the BIND/REBIND Package panel using the Location field on the Package BIND Overrides panel. Use the Package BIND Overrides panel to make global changes to BIND parameters for more than one package or to override attributes of an existing package when generating a BIND or REBIND command.

Specifying the Remote DB2 Subsystem

Overview

This unit tells you how to identify the remote server where processing is to take place.

Background about defining a default remote server

Defining a default remote server is useful only in those cases when you always use the same DB2 subsystem as the remote server. If you plan to perform remote functions on a variety of servers, you will want to identify the remote server as a step in the process you are performing.

Defining the default remote server using a Housekeeping option

You can define a default remote server by using these steps.

Step	Action
1	On the Housekeeping Menu panel, select SET Defaults.
2	On the SET Defaults panel, type the identifier (remote location) of the remote server in the Default Remote Location field.

Identifying the remote server using the SETL command

You can identify the remote server to use by issuing the SETL global command from an object list panel. If you have already specified a default remote location on the SET Defaults Housekeeping panel, use this procedure to override your default specification. The override is in effect for this session only, unless you choose to make your selection permanent and have the authority to do so. Use the steps in the table.

Step	Action
1	Do one of the following: <ul style="list-style-type: none"> On the command line of the object list panel, type the command SETL. On the Global Commands Menu panel, select SETL.
2	Press Enter. Result: !DB/EXPLAIN displays the SET Defaults panel, on which you can identify the remote location to be used.

Specifying the remote server to use for an EXPLAIN

When you request a remote EXPLAIN with statistics gathering from an object list panel (using the GEXPLAR, GEXPLOR, or GEXPLR command or the @ select), !DB/EXPLAIN displays the Statistics Gathering panel. On this panel, enter the name of the server and the amount of statistics to gather. If you do not name a remote server, the server defaults to the local server.

```

----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 1,394
CMD ==>                               SCROLL ==> PAGE
----- STATISTICS GATHERING -----
-- Enter the server from which you wish to retrieve catalog statistics.
-- If no value is entered the current server will be set to local.
SE Server:
-- Amount of data to gather: 2
--
-- 1. - SYSTABLESPACE, SYSTABLES, SYSINDEXES
-- 2. - Data specified in 1 plus
-- SYSCOLUMNS, SYSCOLSTATS, SYSCOLDIST, SYSCOLDISTSTATS
-- 3. - Data specified in 2 plus
-- SYSKEYS
@ 4. - Data specified in 2 plus
-- SYSTABLEPART, SYSTABSTATS, SYSINDEXPART, SYSINDEXSTATS
-- 5. - Data specified in 3 plus
-- SYSTABLEPART, SYSTABSTATS, SYSINDEXPART, SYSINDEXSTATS
--
-- ENTER to process request END to abort request
--
--
--

```

If you requested that no statistics be gathered (EXPLAR, EXPLOR, EXPLR, or #), !DB/EXPLAIN displays a panel requesting entry of the server name. If you do not enter an identifier, the value defaults to the name of the local server. !DB/EXPLAIN performs the requested action and displays the results.

```

----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 1,394
CMD ==>                               SCROLL ==> PAGE
----- NO STATISTICS GATHERING -----
-- Enter the remote server. If no value is entered
-- the current server will be set to LOCAL
SE Server:
--
-- # ENTER to process request END to abort request
--
--
--
----- LOCK LOCK
S INSERTS XCL SHR
-----
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
4 0 0 0
2 0 0 0
4 0 0 0
0 0 2 0
ALECDEV1 4 28 8 0 0
ALECLCK1 1 15 0 2 0

```

Specifying the remote server to use for gathering statistics

When you are on the What-if or Estimator panel, and you want to gather statistics using the GSTATS command, !DB/EXPLAIN displays a panel requesting entry of the name of the server and the amount of statistics to gather. You can enter the identifier of the remote system. If you do not enter an identifier, the server defaults to the local server.

```

----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 1,394
CMD ==> GSTATS                               SCROLL ==> PAGE
----- STATISTICS GATHERING -----
--
-- Enter the server from which you wish to retrieve catalog statistics.
-- If no value is entered the current server will be set to local.
SE  Server:
--  Amount of data to gather: 2
--
-- 1. - SYSTABLESPACE, SYSTABLES, SYSINDEXES
-- 2. - Data specified in 1 plus
--     SYSCOLUMNS, SYSCOLSTATS, SYSCOLDIST, SYSCOLDISTSTATS
-- 3. - Data specified in 2 plus
--     SYSKEYS
-- 4. - Data specified in 2 plus
--     SYSTABLEPART, SYSTABSTATS, SYSINDEXPART, SYSINDEXSTATS
-- 5. - Data specified in 3 plus
--     SYSTABLEPART, SYSTABSTATS, SYSINDEXPART, SYSINDEXSTATS
--
-- ENTER to process request   END to abort request
--
--
--

```

Specifying the remote server to use for updating statistics

When you are on the What-if or Estimator panel, and you want to update catalog statistics using the UPDT command, !DB/EXPLAIN displays a panel requesting entry of the name of the server on which catalog statistics are to be updated. You can enter the identifier of the remote system. If you do not enter an identifier, the server defaults to the local server.

```

----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 12
CMD ==> UPDT                               SCROLL ==> PAGE
----- STATISTICS UPDATE -----
--
-- Enter the server on which you wish to update catalog statistics.
-- If no value is entered the current server will be set to local.
--
-- Server:
--
-- ENTER to process update request   END to abort update request
--
--
--

```

Working With a Remote EXPLAIN

Overview

This unit describes what you can do after you have requested a remote EXPLAIN.

Background about the display of a remote EXPLAIN

When you have requested that an EXPLAIN be performed on a remote DB2 subsystem, the results of the EXPLAIN are displayed to you at your local terminal. However, *the results are not stored in the !DB/EXPLAIN plan table extract either at the remote DB2 subsystem or at the local subsystem on which they are displayed.* When you end the display, either by typing **END** on the command line or by pressing the appropriate function key, the results of the remote EXPLAIN are lost.

Saving the results of the remote EXPLAIN

Use these steps to save the results of the remote EXPLAIN in the user PDS.

Step	Action
1	<p>On the EXPLAIN display, type one of the following on the command line:</p> <ul style="list-style-type: none"> ● EXODS ● EXODS <i>membername</i>
2	<p>Press Enter.</p> <p>Result: One of these actions occurs:</p> <ul style="list-style-type: none"> ● If you typed EXODS on the command line and you specified on the Housekeeping Output Options Defaults panel that the Output Options panel should be displayed on an EXPLAIN, !DB/EXPLAIN displays the Output Options panel on which you can specify the name of the member in which to store the results of the remote EXPLAIN. ● If you typed EXODS on the command line and you <i>did not</i> specify on the Housekeeping Output Options Defaults panel that the Output Options panel should be displayed on an EXPLAIN, !DB/EXPLAIN stores the results of the remote EXPLAIN in the user PDS under the default member name. If a member with that name already exists, !DB/EXPLAIN replaces it. ● If you typed EXODS <i>membername</i> on the command line, !DB/EXPLAIN stores the results of the remote EXPLAIN in the member you specify. If a member already exists with the name you specify, !DB/EXPLAIN replaces it.

Some Usage Scenarios

Overview

This unit provides some scenarios that can suggest ways you can use !DB/EXPLAIN's remote functions.

Background about the scenarios

In the scenarios, you have a test system, DB2T, and a production system, DB2P. You want to EXPLAIN several application plans on your production system. You then want to play some 'what-if' games to determine how these plans might run more effectively. You do not want to impact your production system. Thus, you will want to make your test system look like your production system by altering its catalog statistics.

Effect of permanently updating the test system catalog

By making catalog statistics on your test system equivalent to those on your production system, all subsequent *EXPL* and WIF* commands (for example EXPLAR, GEXPLR, WIFR) will run on the test system as they would on the production system. This permits you to review access path information and evaluate your SQL code without impacting your production environment.

Example to temporarily update the test system catalog and run an EXPLAIN

Use these steps to temporarily update the test system catalog and run an EXPLAIN.

Step	Action
1	Enter the W (what-if) select from your object list (plans, DBRMs, packages, or statements) panel on test system DB2T.
2	Type GSTATS on the command line. When !DB/EXPLAIN displays the Statistics Gathering panel, type DB2P in the Server field. Also specify the amount of statistics to gather.
3	When you receive a message that statistics have been gathered, type WIFL on the command line. This causes !DB/EXPLAIN to temporarily update the DB2T catalog with the statistics from DB2P and perform an EXPLAIN.

Example to permanently update the test system catalog

Use these steps to permanently update the test system catalog. (To ensure that your update is made to the actual catalog and not to the shadow catalog, you must specify **Y** for the Use Real Catalog Tables for Update field on the Miscellaneous Defaults Housekeeping panel.)

Step	Action
1	On test system DB2T, type GSTATS on the command line of a What-if panel. When !DB/EXPLAIN displays the Statistics Gathering panel, type DB2P in the Server field and also specify the amount of statistics to gather.
2	When you receive a message that statistics have been gathered, type UPDTL on the command line. This causes !DB/EXPLAIN to update the local (DB2T) catalog with the statistics from the remote system (DB2P).

Performing Remote Functions in Batch

Overview

This unit discusses the use of !DB/EXPLAIN remote functions in batch and provides examples. The section “Using Batch Facilities” on page 399 is prerequisite to using this unit.

Performing a remote EXPLAIN in batch

You can perform a remote EXPLAIN in batch using the commands in the example below. In this example, the remote DB2 subsystem on which the EXPLAIN is performed is the one you have defined on the Housekeeping SET Defaults panel. If you have not defined a remote subsystem, !DB/EXPLAIN performs the EXPLAIN at the local DB2 subsystem. The example assumes you have already filtered the package list so that you are EXPLAINing only those packages you are interested in.

```
//SYSIN DD *
KA      (Identify packages as object to use)
EXPLAR  (Perform a remote EXPLAIN; do not gather stats)
EXODS   (Output the EXPLAIN to your user PDS)
END     (Terminate processing)
```

Performing a batch EXPLAIN on a remote subsystem you specify

To perform the remote EXPLAIN on a DB2 subsystem other than the one you have identified on the SET Defaults panel, or to override your specification on the SET Defaults panel, use this example:

```
//KTECTL DD *
EXP_DFLT_REMOTE=DB2P (Set remote subsystem ID to DB2P)
//SYSIN DD *
KA      (Identify pkgs. as object to use)
GEXPLAR (Remote EXPLAIN; gather stats)
EXODS mymember (Output EXPLAIN to member mymember)
END     (Terminate processing)
```

Performing a remote batch EXPLAIN for an application module

This example defines the remote subsystem to be used when EXPLAINing the SQL statements imbedded in a module of application code. For a more complete illustration of the JCL to perform this function, review member KTEBSQLS in the *hilev.CNTL* library distributed with !DB/EXPLAIN.

```
//JOB CARD
//KTEBSQLS PROC DB2LOAD=,...
:
//EXBATCH EXEC PGM=IKJEFT1B,...
:
//EXBATCH.KTECNTL DD *
EXP_DFLT_REMOTX=DB2P (Set remote subsystem ID to DB2P)
//EXBATCH.SYSIN DD *
GEXPLAR (Remote EXPLAIN; gather stats)
EXODS mymember (Output EXPLAIN to member mymember)
END (Terminate processing)
```

Performing !DB/EXPLAIN Tasks

Chapter 15. Example Tasks

Introduction

This chapter guides you through scenarios of common tasks you can perform with !DB/EXPLAIN. An overview of each task precedes each scenario. Tasks are broken into easy-to-follow steps which include figures.

Chapter contents

Accessing !DB/EXPLAIN Using the KTEON EXEC	240
Accessing !DB/EXPLAIN Using the KTEON EXEC to Generate JCL	244
Execute an SQL Statement and Review its Results	252
Analyze a High-Cost Package SQL Query Using the Costs Panel	256
Print the Results of a Cost Analysis	264
Execute a “Merge Since” Extract	266
EXPLAIN SQL Statements in a Source Program and Review the Results	267
Set Up General Housekeeping Options to Customize !DB/EXPLAIN	275
Convert a Plan DBRM to a Package Using the Z Select . . .	281
Convert Multiple Plan DBRMs to Packages Using the BINDCNV Command	285
Convert DBRMs in Multiple Plans to Packages Using the BINDCNV Command	290
Change All Plans with an Isolation Level of RR (Repeatable Read) to CS (Cursor Stability)	295

Accessing !DB/EXPLAIN Using the KTEON EXEC

Overview

In this scenario, you will use the Online Menu to access !DB/EXPLAIN. In the example, TDDB00.SQL is a partitioned dataset.

Process to use the Online Menu

To use !DB/EXPLAIN's Online Menu, follow the steps in the table.

Step	Action
1	From the !DB/Tools Product Selection Menu, type 5 to select the !DB/EXPLAIN Online Menu
2	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Online Menu.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==>> O N L I N E M E N U Extract ID ==>> Estae Flag ==>> Y Create Batch JCL? ==>> N (Y Yes N No) Specify Additional Data Sets? ==>> N (Y Yes N No) Specify Member Lists? ==>> N (Y Yes N No) Option ==>> 1 1 SQL Statements Data Set Name : File Format : 2 BIND Commands Data Set Name : Allocate DBRMLIB?: (Y Yes N No E Edit DBRMLIB concatenation) 3 DBRM Data Set Name : Owner/Qualifier : Plan: ----- (C) Copyright CANDLE Corp. 1993 - 1999 ----- </pre> </div>

Process to use the Online Menu (continued)

Step	Action
3	<p>On the Online Menu, type N in the Create Batch JCL? field. Type the data set name 'TDDB00.SQL' in the Data Set Name field and S in the File Format field.</p> <pre data-bbox="573 474 1248 921"> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> O N L I N E M E N U Extract ID ==> D31A Estae Flag ==> Y Create Batch JCL? ==> N (Y Yes N No) Specify Additional Data Sets? ==> N (Y Yes N No) Specify Member Lists? ==> N (Y Yes N No) Option ==> 1 1 SQL Statements Data Set Name : TDDB00.SQL File Format : S 2 BIND Commands Data Set Name : ATtlocate DBRMLIB?: (Y Yes N No E Edit DBRMLIB concatenation) 3 DBRM Data Set Name : Owner/Qualifier : Plan: ----- (C) Copyright CANDLE Corp. 1993 - 1999----- </pre>
4	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays a list of members belonging to the partitioned data set TDDB00.SQL.</p>
5	<p>Select the members you want to process by typing S next to each name you want to select.</p> <pre data-bbox="573 1178 1248 1556"> Select SQL Members -- TDDB00.SQL ----- 00001 OF 00015 Command ==> Scroll ==> CSR M E M B E R L I S T Cmds: DO (Menu) Selects: ? (Menu) ----- NAME VV.MM CREATED CHANGED SIZE INIT MOD ID SQLXAB 01.02 99/06/10 99/06/10 17:51 20 6 0 TDDB00A S SQLXCAN 01.07 99/08/17 99/06/20 02:12 8 174 0 TDDB00 SQLXCANA 01.00 99/09/29 99/09/29 01:06 8 8 0 TDDB00 SQLXCANB S SQLXCANS 01.00 99/08/25 99/08/25 08:03 33 33 0 TDDB00 SQLXCANK 01.00 99/10/08 99/10/08 04:31 50 50 0 TDDB00A SQLXCAN1 01.00 99/12/09 99/12/09 10:34 3 3 0 TDDB00A SQLXCAN2 01.04 99/08/25 99/03/29 00:36 489 387 0 TDDB00A SQLXCAN3 01.04 99/09/14 99/08/24 03:58 1054 427 0 TDDB00A SQLXCAN5 01.00 99/09/23 99/09/23 05:50 3 3 0 TDDB00 SQLXCAN9 01.03 99/09/28 99/08/25 04:27 455 447 0 TDDB00 SQLXHEAD 01.00 99/11/09 99/11/09 11:50 14 14 0 TDDB00 SQLXOUT 01.00 99/08/19 99/08/19 23:39 243 243 0 TDDB00A </pre>

Process to use the Online Menu (continued)

Step	Action
6	<p>After you have selected all of the members you want to process, Press Enter.</p> <p>Result: !DB/EXPLAIN displays the list of members showing the members you selected.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> Select SQL Members -- TDDB00.SQL ----- 00001 OF 00015 Command ==>> Scroll ==>> CSR M E M B E R L I S T Cmds: DO (Menu) Selects: ? (Menu) ----- NAME VV.MM CREATED CHANGED SIZE INIT MOD ID SQLXAB 01.02 99/06/10 99/06/10 17:51 20 6 0 TDDB00A SQLXCAN *SELECTD 01.07 99/08/17 99/06/20 02:12 8 174 0 TDDB00 SQLXCANA 01.00 99/09/29 99/09/29 01:06 8 8 0 TDDB00 SQLXCANB SQLXCANS *SELECTD 01.00 99/08/25 99/08/25 08:03 33 33 0 TDDB00 SQLXCANK 01.00 99/10/08 99/10/08 04:31 50 50 0 TDDB00A SQLXCAN1 01.00 99/12/09 99/12/09 10:34 3 3 0 TDDB00A SQLXCAN2 01.04 99/08/25 99/03/29 00:36 489 387 0 TDDB00A SQLXCAN3 01.04 99/09/14 99/08/24 03:58 1054 427 0 TDDB00A SQLXCAN5 01.00 99/09/23 99/09/23 05:50 3 3 0 TDDB00 SQLXCAN9 01.03 99/09/28 99/08/25 04:27 455 447 0 TDDB00 SQLXHEAD 01.00 99/11/09 99/11/09 11:50 14 14 0 TDDB00 SQLXOUT 01.00 99/08/19 99/08/19 23:39 243 243 0 TDDB00A </pre> </div>

Process to use the Online Menu (continued)

Step	Action
7	<p>Type END on the command line or press the appropriate function key.</p> <p>Result: This example assumes you have specified N to the Bypass Statements Display? field on Housekeeping's SQL Defaults panel. !DB/EXPLAIN displays the Statements panel containing statements for each of the members you selected.</p> <div data-bbox="571 592 1248 1058" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D42B ----- LINE 1 OF 13 Cmd ==> SCROLL ==> CSR S T A T E M E N T S Cmds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- STMT S I C SEL NO T S H SQL TEXT ----- PACKAGE=DSNHYCRD COLLID=DSNHYCRD CONTOKEN=155403E608555E10 PRECOMPILE=1999/02/13 07:38:37.843832 - 1051 N DECLARE CURSOR1 CURSOR FOR - 1056 N DECLARE CURSOR2 CURSOR FOR - 204 N SELECT TYPE , COLCOUNT - 277 N OPEN CURSOR1 - 278 N FETCH CURSOR1 - 360 N FETCH CURSOR1 - 375 N CLOSE CURSOR1 - 420 N OPEN CURSOR2 - 421 N FETCH CURSOR2 - 637 N FETCH CURSOR2 - 652 N CLOSE CURSOR2 ***** BOTTOM OF LIST ***** </pre> </div>
8	<p>To process BIND commands or DBRMs, follow steps 1 through 6 above, substituting the appropriate function code in the Function field of the Online Menu. !DB/EXPLAIN displays member lists for your selections that are appropriate to the type of object being processed. In each case, when the Statements panel is displayed, it contains statements for all of the members you have selected from your member list.</p>

Accessing !DB/EXPLAIN Using the KTEON EXEC to Generate JCL

Overview

In this scenario, you will use the Online Menu to generate batch JCL. In the example, TDDB00.SQL is a partitioned dataset.

Process to use the Online Menu to generate batch JCL.

To generate batch JCL using !DB/EXPLAIN's Online Menu, follow the steps in the table.

Step	Action
1	From the !DB/Tools Product Selection Menu, type 5 to select the !DB/EXPLAIN Online Menu.
2	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Online Menu.</p> <pre data-bbox="573 919 1247 1360"> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> O N L I N E M E N U Extract ID ==> Estae Flag ==> Y Create Batch JCL? ==> N (Y Yes N No) Specify Additional Data Sets? ==> N (Y Yes N No) Specify Member Lists? ==> N (Y Yes N No) Option ==> 1 1 SQL Statements Data Set Name : File Format : 2 BIND Commands Data Set Name : Allocate DBRMLIB?: (Y Yes N No E Edit DBRMLIB concatenation) 3 DBRM Data Set Name : Owner/Qualifier : Plan: ----- (C) Copyright CANDLE Corp. 1993 - 1999 ----- </pre>

Process to use the Online Menu to generate batch JCL (continued)

Step	Action
3	<p>On the Online Menu, type:</p> <ul style="list-style-type: none"> ● D31A in the Extract ID field ● Y in the Create Batch JCL? field to indicate a batch job ● 1 to select EXPLAIN for SQL statements in the Function field ● The dataset name 'TDDB00.SQL' in the dataset Name field ● S in the File Format field <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==>> O N L I N E M E N U Extract ID ==>> D31A Estae Flag ==>> Y Create Batch JCL? ==>> Y (Y Yes N No) Specify Additional Data Sets? ==>> N (Y Yes N No) Specify Member Lists? ==>> N (Y Yes N No) Option ==>> 1 1 SQL Statements Data Set Name : TDDB00.SQL File Format : S 2 BIND Commands Data Set Name : Allocate DBRMLIB?: (Y Yes N No E Edit DBRMLIB concatenation) 3 DBRM Data Set Name : Owner/Qualifier : Plan: ----- (C) Copyright CANDLE Corp. 1993 - 1999 ----- </pre> </div>
4	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays a list of members belonging to the partitioned dataset TDDB00.SQL.</p>

Process to use the Online Menu to generate batch JCL (continued)

Step	Action																																																																																																																
5	<p>Select the members you want to process by typing S next to each name you want to select.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre>Select SQL Members -- TDDB00.SQL ----- 00001 OF 00015 Command ==>> Scroll ==>> CSR M E M B E R L I S T Cmds: DO (Menu) Selects: ? (Menu) -----</pre> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>NAME</th> <th>VV.MM</th> <th>CREATED</th> <th>CHANGED</th> <th>SIZE</th> <th>INIT</th> <th>MOD</th> <th>ID</th> </tr> </thead> <tbody> <tr><td>SQLXAB</td><td>01.02</td><td>99/06/10</td><td>99/06/10 17:51</td><td>20</td><td>6</td><td>0</td><td>TDDB00A</td></tr> <tr><td>S SQLXCAN</td><td>01.07</td><td>99/08/17</td><td>99/06/20 02:12</td><td>8</td><td>174</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXCANA</td><td>01.00</td><td>99/09/29</td><td>99/09/29 01:06</td><td>8</td><td>8</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXCANB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>S SQLXCANS</td><td>01.00</td><td>99/08/25</td><td>99/08/25 08:03</td><td>33</td><td>33</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXCANX</td><td>01.00</td><td>99/10/08</td><td>99/10/08 04:31</td><td>50</td><td>50</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN1</td><td>01.00</td><td>99/12/09</td><td>99/12/09 10:34</td><td>3</td><td>3</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN2</td><td>01.04</td><td>99/08/25</td><td>99/03/29 00:36</td><td>489</td><td>387</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN3</td><td>01.04</td><td>99/09/14</td><td>99/08/24 03:58</td><td>1054</td><td>427</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN5</td><td>01.00</td><td>99/09/23</td><td>99/09/23 05:50</td><td>3</td><td>3</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXCAN9</td><td>01.03</td><td>99/09/28</td><td>99/08/25 04:27</td><td>455</td><td>447</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXHEAD</td><td>01.00</td><td>99/11/09</td><td>99/11/09 11:50</td><td>14</td><td>14</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXOUT</td><td>01.00</td><td>99/08/19</td><td>99/08/19 23:39</td><td>243</td><td>243</td><td>0</td><td>TDDB00A</td></tr> </tbody> </table> </div>	NAME	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	ID	SQLXAB	01.02	99/06/10	99/06/10 17:51	20	6	0	TDDB00A	S SQLXCAN	01.07	99/08/17	99/06/20 02:12	8	174	0	TDDB00	SQLXCANA	01.00	99/09/29	99/09/29 01:06	8	8	0	TDDB00	SQLXCANB								S SQLXCANS	01.00	99/08/25	99/08/25 08:03	33	33	0	TDDB00	SQLXCANX	01.00	99/10/08	99/10/08 04:31	50	50	0	TDDB00A	SQLXCAN1	01.00	99/12/09	99/12/09 10:34	3	3	0	TDDB00A	SQLXCAN2	01.04	99/08/25	99/03/29 00:36	489	387	0	TDDB00A	SQLXCAN3	01.04	99/09/14	99/08/24 03:58	1054	427	0	TDDB00A	SQLXCAN5	01.00	99/09/23	99/09/23 05:50	3	3	0	TDDB00	SQLXCAN9	01.03	99/09/28	99/08/25 04:27	455	447	0	TDDB00	SQLXHEAD	01.00	99/11/09	99/11/09 11:50	14	14	0	TDDB00	SQLXOUT	01.00	99/08/19	99/08/19 23:39	243	243	0	TDDB00A
NAME	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	ID																																																																																																										
SQLXAB	01.02	99/06/10	99/06/10 17:51	20	6	0	TDDB00A																																																																																																										
S SQLXCAN	01.07	99/08/17	99/06/20 02:12	8	174	0	TDDB00																																																																																																										
SQLXCANA	01.00	99/09/29	99/09/29 01:06	8	8	0	TDDB00																																																																																																										
SQLXCANB																																																																																																																	
S SQLXCANS	01.00	99/08/25	99/08/25 08:03	33	33	0	TDDB00																																																																																																										
SQLXCANX	01.00	99/10/08	99/10/08 04:31	50	50	0	TDDB00A																																																																																																										
SQLXCAN1	01.00	99/12/09	99/12/09 10:34	3	3	0	TDDB00A																																																																																																										
SQLXCAN2	01.04	99/08/25	99/03/29 00:36	489	387	0	TDDB00A																																																																																																										
SQLXCAN3	01.04	99/09/14	99/08/24 03:58	1054	427	0	TDDB00A																																																																																																										
SQLXCAN5	01.00	99/09/23	99/09/23 05:50	3	3	0	TDDB00																																																																																																										
SQLXCAN9	01.03	99/09/28	99/08/25 04:27	455	447	0	TDDB00																																																																																																										
SQLXHEAD	01.00	99/11/09	99/11/09 11:50	14	14	0	TDDB00																																																																																																										
SQLXOUT	01.00	99/08/19	99/08/19 23:39	243	243	0	TDDB00A																																																																																																										
6	<p>After you have selected all of the members you want to process, Press Enter.</p> <p>Result: !DB/EXPLAIN displays the list of members showing the members you selected.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre>Select SQL Members -- TDDB00.SQL ----- 00001 OF 00015 Command ==>> Scroll ==>> CSR M E M B E R L I S T Cmds: DO (Menu) Selects: ? (Menu) -----</pre> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>NAME</th> <th>VV.MM</th> <th>CREATED</th> <th>CHANGED</th> <th>SIZE</th> <th>INIT</th> <th>MOD</th> <th>ID</th> </tr> </thead> <tbody> <tr><td>SQLXAB</td><td>01.02</td><td>99/06/10</td><td>99/06/10 17:51</td><td>20</td><td>6</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN *SELECTD</td><td>01.07</td><td>99/08/17</td><td>99/06/20 02:12</td><td>8</td><td>174</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXCANA</td><td>01.00</td><td>99/09/29</td><td>99/09/29 01:06</td><td>8</td><td>8</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXCANB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>SQLXCANS *SELECTD</td><td>01.00</td><td>99/08/25</td><td>99/08/25 08:03</td><td>33</td><td>33</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXCANX</td><td>01.00</td><td>99/10/08</td><td>99/10/08 04:31</td><td>50</td><td>50</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN1</td><td>01.00</td><td>99/12/09</td><td>99/12/09 10:34</td><td>3</td><td>3</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN2</td><td>01.04</td><td>99/08/25</td><td>99/03/29 00:36</td><td>489</td><td>387</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN3</td><td>01.04</td><td>99/09/14</td><td>99/08/24 03:58</td><td>1054</td><td>427</td><td>0</td><td>TDDB00A</td></tr> <tr><td>SQLXCAN5</td><td>01.00</td><td>99/09/23</td><td>99/09/23 05:50</td><td>3</td><td>3</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXCAN9</td><td>01.03</td><td>99/09/28</td><td>99/08/25 04:27</td><td>455</td><td>447</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXHEAD</td><td>01.00</td><td>99/11/09</td><td>99/11/09 11:50</td><td>14</td><td>14</td><td>0</td><td>TDDB00</td></tr> <tr><td>SQLXOUT</td><td>01.00</td><td>99/08/19</td><td>99/08/19 23:39</td><td>243</td><td>243</td><td>0</td><td>TDDB00A</td></tr> </tbody> </table> </div>	NAME	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	ID	SQLXAB	01.02	99/06/10	99/06/10 17:51	20	6	0	TDDB00A	SQLXCAN *SELECTD	01.07	99/08/17	99/06/20 02:12	8	174	0	TDDB00	SQLXCANA	01.00	99/09/29	99/09/29 01:06	8	8	0	TDDB00	SQLXCANB								SQLXCANS *SELECTD	01.00	99/08/25	99/08/25 08:03	33	33	0	TDDB00	SQLXCANX	01.00	99/10/08	99/10/08 04:31	50	50	0	TDDB00A	SQLXCAN1	01.00	99/12/09	99/12/09 10:34	3	3	0	TDDB00A	SQLXCAN2	01.04	99/08/25	99/03/29 00:36	489	387	0	TDDB00A	SQLXCAN3	01.04	99/09/14	99/08/24 03:58	1054	427	0	TDDB00A	SQLXCAN5	01.00	99/09/23	99/09/23 05:50	3	3	0	TDDB00	SQLXCAN9	01.03	99/09/28	99/08/25 04:27	455	447	0	TDDB00	SQLXHEAD	01.00	99/11/09	99/11/09 11:50	14	14	0	TDDB00	SQLXOUT	01.00	99/08/19	99/08/19 23:39	243	243	0	TDDB00A
NAME	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	ID																																																																																																										
SQLXAB	01.02	99/06/10	99/06/10 17:51	20	6	0	TDDB00A																																																																																																										
SQLXCAN *SELECTD	01.07	99/08/17	99/06/20 02:12	8	174	0	TDDB00																																																																																																										
SQLXCANA	01.00	99/09/29	99/09/29 01:06	8	8	0	TDDB00																																																																																																										
SQLXCANB																																																																																																																	
SQLXCANS *SELECTD	01.00	99/08/25	99/08/25 08:03	33	33	0	TDDB00																																																																																																										
SQLXCANX	01.00	99/10/08	99/10/08 04:31	50	50	0	TDDB00A																																																																																																										
SQLXCAN1	01.00	99/12/09	99/12/09 10:34	3	3	0	TDDB00A																																																																																																										
SQLXCAN2	01.04	99/08/25	99/03/29 00:36	489	387	0	TDDB00A																																																																																																										
SQLXCAN3	01.04	99/09/14	99/08/24 03:58	1054	427	0	TDDB00A																																																																																																										
SQLXCAN5	01.00	99/09/23	99/09/23 05:50	3	3	0	TDDB00																																																																																																										
SQLXCAN9	01.03	99/09/28	99/08/25 04:27	455	447	0	TDDB00																																																																																																										
SQLXHEAD	01.00	99/11/09	99/11/09 11:50	14	14	0	TDDB00																																																																																																										
SQLXOUT	01.00	99/08/19	99/08/19 23:39	243	243	0	TDDB00A																																																																																																										

Process to use the Online Menu to generate batch JCL (continued)

Step	Action
7	<p>If the list is correct, type END on the command line or press the appropriate function key.</p> <p>Result: !DB/EXPLAIN displays the Generate SQL JCL panel. On this panel:</p> <ul style="list-style-type: none"> • In the Edit Commands (SYSIN)? field, specify Y to make any desired changes to batch commands in your job. • In the Edit Options (KTECNTL)? field, specify Y to make any desired changes to the KTECNTL statements in your job. Specifying Y here permits you to specify overrides for profile variables. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> GENERATE JCL FOR PROCESSING SQL STATEMENTS Extract ID ==> D31A Data Set Name ==> TDDB00.USER.SQL(SQL) File Format ==> S Specify Additional Data Sets? ==> N (Y Yes N No) Specify Member Lists? ==> N (Y Yes N No) Select Scenario? ==> N (Y Yes N No) Initial Command? ==> Y (Y Yes N No) Format Display? ==> N (Y Yes N No C Current) Filter Display? ==> N (Y Yes N No C Current) Print Display? ==> N (Y Yes N No) Specify Header Lines? ==> N (Y Yes N No) Edit Commands (SYSIN)? ==> Y (Y Yes N No) Edit Options (KTECNTL)? ==> Y (Y Yes N No R Replace) Options Member ==> &USER Q Change JOB Information? ==> N (Y Yes N No) ----- (C) Copyright CANDLE Corp. 1993 - 1999 ----- </pre> </div>

Process to use the Online Menu to generate batch JCL (continued)

Step	Action
8	<p>If you specified Y for Change SYSIN?, this panel is displayed to permit you to specify the commands you want to execute. On the illustrative panel below, the commands cause !DB/EXPLAIN to EXPLAIN the SQL statements and gather catalog statistics. (If you do not want to gather statistics, substitute the command EXPLA for the command GEXPLA.)</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- COLS 001 072 Cmd ==> Scroll ==> PAGE EDIT SYSIN - TDD800.KTEBSQLS.KTESYSIS Cmds: END Exit CANCEL Cancel KTEMODEL Copy sample SYSIN statements. (Use with A or B line command.) KTESYSIS Browse sample SYSIN statements. ***** ***** TOP OF DATA ***** 000001 * 000002 * DB/EXPLAIN commands. See CNTL(KTESYSIS) for more info. 000003 * 000004 GEXPLA 000005 PRNT 000006 END ***** ***** BOTTOM OF DATA ***** </pre> </div>
9	<p>Next, do one of the following:</p> <ul style="list-style-type: none"> ● Type END on the command line or press the appropriate function key. ● Type KTESYSIS on the command line to view example sets of !DB/EXPLAIN commands. ● Type KTEMODEL on the command line to copy example sets of !DB/EXPLAIN commands to the JCL you are creating. <p>For additional information about KTESYSISIN and KTESYSIS, see “Control members shipped with !DB/EXPLAIN” on page 405</p>

Process to use the Online Menu to generate batch JCL (continued)

Step	Action
10	<p>Press Enter.</p> <p>Result: If you specified Y for Change KTECNTL? !DB/EXPLAIN displays the Edit KTECNTL Deck panel on which you can do one of the following:</p> <ul style="list-style-type: none"> ● Type END or press the appropriate function key to exit the panel. ● Type KTECNTL on the command line to browse the member KTECNTL shipped in the !DB/EXPLAIN <i>hilevCNTL</i> library. If you want to override any of the members in KTECNTL, type the appropriate override statement as shown in the display. If you don't know what variable to use to override a variable in KTECNTL, browse KTEPFVAR. ● Type KTEPFVAR on the command line to browse the member KTEPFVAR shipped in the !DB/EXPLAIN <i>hilevCNTL</i> library. KTEPFVAR contains a list of the profile variables and their definitions. <p>For additional information about KTECNTL and KTEPFVAR, see “Control members shipped with !DB/EXPLAIN” on page 405</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- COLS 001 072 Cmd ==> Scroll ==> PAGE EDIT KTECNTL - TDT00.KTEBSQLS.KTECNTL2 Cmds: END Exit CANCEL Cancel KTECNTL Browse sample KTECNTL statements. KTEPFVAR Browse profile variable definitions. ***** ***** TOP OF DATA ***** 000001 * 000002 * Overrides for profile variables. See CNTL(KTECNTL) for more info. 000003 * 000004 EXP_RETAIN_DISP=N ***** ***** BOTTOM OF DATA ***** </pre> </div>

Process to use the Online Menu to generate batch JCL (continued)

Step	Action
11	<p>When you end your work on the Edit KTECNTL panel, !DB/EXPLAIN displays the Output Options panel to permit you to specify the library where the JCL is to be stored.</p> <div data-bbox="571 474 1250 940" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A----- Cmd ==> O U T P U T O P T I O N S You have requested a function which will output data to a PDS member. Specify the PDS, member name and whether to modify onto member or not. Library: TDKT.DBTOOLS.V311.EXP.USER Member : JCL Modify onto member: N ENTER to accept values END to cancel request </pre> </div>
12	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Output JCL Options panel. To review the JCL, type 2 on the command line.</p> <div data-bbox="571 1115 1250 1528" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> O U T P U T J C L O P T I O N S The JCL has been saved in TDKT.DBTOOLS.V311.EXP.USER(JCL). Specify next action to be performed. Select Action ==> 1. SUBMIT Submit the member. 2. EDIT Edit the member. 3. DELETE Delete the member (and exit). 4. PRINT Print the member. 5. DO LATER Changes are saved, exit now. </pre> </div>
13	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the JCL which you can edit and submit. To submit, type SUB on the command line and press ENTER.</p>

Generated JCL

This is an example of the JCL generated as a result of the steps in this unit.

```

//TDDB00A JOB (** ACCT INFO **),'DB/EXPLAIN BATCH',
//  MSGLEVEL=(1,1),MSGCLASS=A,CLASS=A,NOTIFY=TDDB00
/**
/**=====
/**===
/**=== Purpose: Execute !DB/Explain for SQL file in batch ===
/**=== (Batch equivalent of KTEXPLA). ===
/**===
/**=====
/**
//KTEBSQLS EXEC PGM=IKJEFT1B,DYNAMNBR=20,REGION=4M
//STEPLIB DD DISP=SHR,DSN=TDDB.DBTOOLS.LOAD
// DD DISP=SHR,DSN=PP.DB2.V3R1.SDSNLOAD
//SYSPROC DD DISP=SHR,DSN=TDDB.DBTOOLS.CLIST
// DD DISP=SHR,DSN=TDDB.DB2.V2R3.DSNCLIST
// DD DISP=SHR,DSN=SYS1.ISPF.CLIST
//SFXMLIB DD DISP=SHR,DSN=TDDB.DBTOOLS.MSGS
//KTCPROFI DD DISP=SHR,DSN=TDDB00A.DBTOOLS.PROFILE
//KTCPROFO DD DISP=SHR,DSN=TDDB00A.DBTOOLS.PROFILE
//SYSTEM DD DUMMY
//SQLPUNCH DD DUMMY
//SYSPUNCH DD DUMMY
//KTCTRACE DD SYSOUT=*
//KTEXCPTS DD SYSOUT=*
//KTESNAP DD SYSOUT=*,OUTLIM=16777215
//KTESTRAC DD SYSOUT=*,OUTLIM=16777215,RECFM=FB,LRECL=121
//SYSERROR DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//REPORTS DD SYSOUT=X
//SYSYSIN DD *
%KTEXPLAB +
    FORMAT(S) +
    LOAD(TDDB.DBTOOLS.LOAD) +
    DB2(D31A) +
    DSN(TDDB00A.CNTL) +
    INCLUDE( +
        SQLXCAN SQLXCANX +
        ) +
    EXCLUDE() +
    VIO(VIO) +
    BLKSIZE(3200) +
    ESTAE(Y)
/**
//KTECNTL DD *
*
* Overrides for profile variables. See CNTL(KTECNTL) for more info.
*
EXP_RETAIN_DISP=N
/**
/**
/** !DB/Explain commands. See CNTL(KTESYSIS) for more info.
/**
//SYSIN DD *
*
* !DB/Explain commands. See CNTL(KTESYSIS) for more info.
*
*
GEXPLA
PRNT
END
/**

```

Execute an SQL Statement and Review its Results

Overview

In this scenario you will

- execute an SQL statement
- review the results of the SQL statement
- review the Statement Execution Summary

Process to execute an SQL statement and review its results

Step	Action
1	<p>To execute an SQL statement, access the Statements panel from the DBRMS, Packages, Plans, or Tables panels. In this example, we will start at the Plans panel.</p> <p>Type S in the Select column.</p> <div data-bbox="573 886 1248 1150" style="border: 1px solid black; padding: 5px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE38 OF 60 CMD ==> SCROLL = ==> PAGE P L A N S Cmds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL PLAN COUNT STMTS SELECTS DELETES UPDATES INSERTS LOCK LOCK ----- S KTE 44 536 8 0 16 0 0 0 - KTEPLAN 44 536 8 0 16 0 0 0 - QMF240 18 117 2 3 3 5 0 0 </pre> </div>
2	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Statements panel. Type SHOW on the command line.</p>

Process to execute an SQL statement and review its results (continued)

Step	Action
3	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN redisplay the Statements panel with expanded statements.</p> <p>Type Z in the Select column to execute the SQL statement.</p> <div data-bbox="573 541 1250 1003" style="border: 1px solid black; padding: 5px;"> <pre> ----- DB/EXPLAIN DB2=D42B ----- LINE 1 OF 13 Cmd ==>> SCROLL ==>> CSR S T A T E M E N T S ----- Cnds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- STMT S I C SEL NO T S H SQL TEXT ----- PACKAGE=DSNHYCRD COLLID=DSNHYCRD CONTOKEN=155403E6085555E10 PRECOMPILE=1999/02/13 07:38:37.843832 - 1051 N DECLARE CURSOR1 CURSOR FOR - 1056 N DECLARE CURSOR2 CURSOR FOR - 204 N SELECT TYPE , COLCOUNT - 277 N OPEN CURSOR1 - 278 N FETCH CURSOR1 - 360 N FETCH CURSOR1 - 375 N CLOSE CURSOR1 - 420 N OPEN CURSOR2 - 421 N FETCH CURSOR2 - 637 N FETCH CURSOR2 - 652 N CLOSE CURSOR2 ***** BOTTOM OF LIST ***** </pre> </div>
4	<p>Press Enter.</p> <p>If your SQL statement contains host variables, !DB/EXPLAIN displays a Host Variables panel on which you can specify any host variables.</p> <p>The Host Variables panel lists all of the host variables referenced by the SQL statement you selected. At this time, you can enter values and attributes for these host variables before executing the statement. The values you enter may or may not be retained depending on the value you specify on the Host Variables panel for the Retain Host Variables? field. If you specified Y, then the host variable values and attributes are retained across sessions until you explicitly reset or delete them.</p> <p>If you receive this panel, specify the appropriate values and press Enter.</p>

Process to execute an SQL statement and review its results (continued)

Step	Action
5	<p>Whether or not you worked with a Host Variables panel, !DB/EXPLAIN next displays the Statement Execution Results panel. Review the results.</p> <pre> SQL OUTPUT -----DB/EXPLAIN DB2=D31A----- ROW 1 OF 558(F) Cmd ==> Scroll ==> CSR NAME CREATOR Y DBNAME TSNAME DBID OBID 1 PLAN_TABLE TDDB16X T C0BDBD PLANTBLS 263 23 2 PLAN_TABLE TSDB038 T C0BDBD PLANTBLS 263 24 3 PLAN_TABLE TSDB09A T C0BDBD PLANTBLS 263 25 4 HOLIDAYS AA14C00 T DSNDB04 HOLIDAYS 4 9 5 SYSCOLUMNS BATMAN T BATMAN BATMAN 276 4 6 SYSINDEXES BATMAN T BATMAN BATMAN 276 8 7 SYSINDEXPART BATMAN T BATMAN BATMAN 276 7 8 SYSKEYS BATMAN T BATMAN BATMAN 276 9 9 SYSTABLEPART BATMAN T BATMAN BATMAN 276 6 10 SYSTABLES BATMAN T BATMAN BATMAN 276 3 11 SYSTABLESPACE BATMAN T BATMAN BATMAN 276 5 12 HZACCT CANDLE T CANDBASE S2TSACCT 285 14 13 HZACCT_SUM CANDLE T CANDBASE S2TSACSM 285 25 14 HZACCT_SUMDAY CANDLE T CANDBASE S2TSACSM 285 43 15 HZACCT_SUMHOUR CANDLE T CANDBASE S2TSACSM 285 34 16 HZACCT_SUMMONTH CANDLE T CANDBASE S2TSACSM 285 61 17 HZACCT_SUMWEEK CANDLE T CANDBASE S2TSACSM 285 52 18 HZACCTBP CANDLE T CANDBASE S2TSACCT 285 17 19 HZACCTBP_SUM CANDLE T CANDBASE S2TSACSM 285 28 20 HZACCTBP_SUMDAY CANDLE T CANDBASE S2TSACSM 285 46 21 HZACCTBP_SUMHOUR CANDLE T CANDBASE S2TSACSM 285 37 22 HZACCTBP_SUMMONTH CANDLE T CANDBASE S2TSACSM 285 64 23 HZACCTBP_SUMWEEK CANDLE T CANDBASE S2TSACSM 285 55 24 HZACCTDF CANDLE T CANDBASE S2TSACCT 285 20 </pre>
6	<p>Press END.</p> <p>Result: !DB/EXPLAIN displays the Statement Execution Summary panel. Review the statement execution summary results.</p> <pre> -----DB/EXPLAIN DB2=D31A----- CMD ==> SCROLL ==> PAGE STATEMENT EXECUTION SUMMARY ----- Cmds: GLOBAL (Menu) ----- DB2 SUBSYSTEM - D31A COST - 12.2 TIMERONS DEFAULT CREATOR - TBCREATE ELAPSED TIME - 1.18 SECS CURRENT SQLID - CPU TIME - 0.10 SECS CURRENT DEGREE - 1 SERVICE UNITS - 2,855.00 SQLERRD(3) - 0 ROWS RETURNED - 0 (ROWS UPDATED) SELECT CREATOR , NAME FROM SYSTABLES , WHERE NAME = (SELECT TBNAME FROM SYSCOLUMNS) ***** BOTTOM OF LIST ***** </pre>

Process to execute an SQL statement and review its results (continued)

Step	Action
7	Press END. Result: !DB/EXPLAIN returns you to the panel from which you invoked the Statements panel.

Analyze a High-Cost Package SQL Query Using the Costs Panel

Overview

In this scenario you will

- View a listing of all packages and their cost
- Sort the package list by maximum cost in descending order
- Display information on the package with the highest maximum cost
- EXPLAIN the package with the highest maximum cost
- Analyze the EXPLAIN output identifying the high-cost SQL query

Begin at the !DB/EXPLAIN Primary Menu. Remember that your default formats may be different than the ones shown.

Process to analyze a high-cost package SQL query using costs

Step	Action
1	On !DB/EXPLAIN's Primary Menu, view a list of all plans and their costs by typing 7 to select DB2 costs.
2	Press Enter. Result: !DB/EXPLAIN displays the Costs panel.
3	On the Costs panel, type CPKG to produce the package list. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 250 CMD ==> cpkg SCROLL ==> PAGE C O S T S Cmds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- S PLAN -----ACCESS TYPES-----SORTS-----METHOD-- L ADG JDG TS MIS NMIS I1 N MX MI MU SU SJ SO SG SRT MSJ NLJ HBJ LCK LPF SPF ----- - CANPLAN 0 0 1 47 1 0 0 0 0 0 0 0 0 11 0 11 0 0 0 0 30 34 2 - CANPLNX 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 - CHES110 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 - CHEX110 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - DSDPLANZ 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 - DSDV220 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 - DSN8BH31 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - DSN8BP31 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 </pre> </div>

Process to analyze a high-cost package SQL query using costs (continued)

Step	Action
<p>4</p>	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Costs panel package list.</p> <pre data-bbox="573 457 1248 940"> ----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 204 CMD ==> format3 SCROLL ==> PAGE C O S T S Cmds: DO (Menu) GLOBAL (Menu)_ Selects: ? (Menu) ----- S PACKAGE -----ACCESS TYPES-----SORTS-----METHOD-- L ADG JDG TS MIS NMIS I1 N MX MI MU SU SJ SO SG SRT MSJ NLJ HBJ LCK LPF SPF ----- - DSN8HC3 0 - DSN8IC1 0 - DSN8IC2 0 - GABSQL01 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - GABSQL02 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - GABSQL02 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - GABSQL03 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - GABSQL03 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 </pre>
<p>5</p>	<p>When !DB/EXPLAIN displays the Package List panel, you want to view the package costs in format 3. Type FORMAT3 on the ommand line of the Costs package list.</p>
<p>6</p>	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Costs panel in format 3.</p> <pre data-bbox="573 1192 1248 1675"> KTEPCOST ----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 102 CMD ==> C O S T S Cmds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL PACKAGE MAXIMUM AVERAGE TOTAL STMT COST STMT COST STMT COST ----- - DSN8HC3 0.0 0.0 0.0 - DSN8IC1 0.0 0.0 0.0 - DSN8IC2 0.0 0.0 0.0 - GABSQL01 10.6 9.7 29.0 - GABSQL02 10.6 9.7 29.0 - GABSQL02 10.6 9.7 29.0 - GABSQL03 10.6 9.7 29.0 - GABSQL03 10.6 9.7 29.0 - GUL8BC3 1,427.1 1,397.9 2,795.7 - KO2SABP 0.0 0.0 0.0 - KO2SABP 0.0 0.0 0.0 - KO2SABS 0.0 0.0 0.0 - KO2SABS 0.0 0.0 0.0 - KO2SACS 0.0 0.0 0.0 - KO2SACS 0.0 0.0 0.0 - KO2SADD 0.0 0.0 0.0 </pre>

Process to analyze a high-cost package SQL query using costs (continued)

Step	Action
7	Sort the Costs format 3 display by maximum cost in descending order. To do so, on the command line of the Costs panel, type SORT? .
8	Press Enter. Result: !DB/EXPLAIN displays the Costs Sorts Menu.
9	<p>On the Costs Sorts menu:</p> <ul style="list-style-type: none"> ● Type 4 in the Sort Key field. ● Type D in the Order field. <p>This causes the costs to be sorted by maximum costs in descending order.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> CMD ==>> COSTS SORTS SCROLL ==>>> PAGE ----- Sort Key ==>> 4 (Number) Order ==>> D A (Asc) D (Desc) 1. CNAME By Name 2. CPCDT By Precomp Date/Time, Name 3. CEXDT By EXPLAIN Date/Time, Name 4. CMCST By Maximum Statement Cost 5. CACST By Average Statement Cost 6. CTCST By Total Statement Cost 7. C#TSS By # TS Scans 8. C#MIS By # Matching IX Scans 9. C#NMI By # Nonmatching IX Scans 10. C#IIS By # I Fetch Scans 11. C#NS By # IX Scans for IN 12. C#MX By # Multiple IX 13. C#MXS By # IX Scans on IX 14. C#MXI By # MIX Intersections 15. C#MXU By # MIX Unions 16. C#SRU By # Sorts for Unique 17. C#SRJ By # Sorts for Joins 18. C#SRQ By # Sorts for Order By 19. C#SRG By # Sorts for Group By 20. C#SRT By # Sorts 21. C#MSJ By # Merge Scan Joins 22. C#NLJ By # Nested Loop Joins 23. C#HBJ By # Hybrid Joins 24. C#LCK By # Locks 25. C#LPF By # List Prefetches 26. C#SPF By # Seq. Prefetches 27. C#ADG By # Access Degree > 1 28. C#JDG By # Join Degree > 1 ENTER to process END to cancel </pre> </div>

Process to analyze a high-cost package SQL query using costs (continued)

Step	Action
10	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Costs package list again, sorted by maximum costs in descending order.</p>
11	<p>To display information about the package having the highest maximum cost, type K in that package's select field.</p> <div data-bbox="573 573 1248 1056" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 102 CMD ==> SCROLL ==> PAGE C O S T S Cnds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL PACKAGE MAXIMUM AVERAGE TOTAL STMT COST STMT COST STMT COST ----- K TTXEMP1 3,571,336.0 1,551,348.0 6,205,394.0 - KTERSTA3 1,928.7 835.3 5,847.3 - KTERSTA3 1,928.7 835.3 5,847.3 - KTERSTA3 1,928.7 835.3 5,847.3 - KTERSTA3 1,928.7 835.3 5,847.3 - KTERSTA3 1,928.7 835.3 5,847.3 - SQLUCIA0 1,923.8 1,089.6 2,179.3 - GULBBC3 1,427.1 1,397.9 2,795.7 - TTXSR05 1,370.6 1,052.1 6,312.3 - TTXSR04 1,370.6 1,052.1 6,312.3 - TTXSR04 1,370.6 1,052.1 6,312.3 - TTXSR03 1,370.6 1,052.1 6,312.3 - TTXSR03 1,370.6 1,052.1 6,312.3 - TTXSR03 1,370.6 1,052.1 6,312.3 - TTXRUM2 1,370.6 362.3 1,449.3 - TTXRUM1 1,370.6 698.4 1,396.8 - TTXSR02 1,338.9 1,338.9 1,338.9 - TTXSR02 1,338.9 1,338.9 1,338.9 </pre> </div>
12	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays information about the selected package.</p> <div data-bbox="573 1230 1248 1493" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 3 CMD ==> format6 SCROLL ==> PAGE P A C K A G E S Cnds: DO (Menu) GLOBAL (Menu) ? ----- SEL PACKAGE VOVRIVDEQCCMDHRSDD LCK LCK LPDESAPXUOHXELMEGR STMTS SELECTS DELETES UPDATES INSERTSXCL SHR ----- - TTXEMP1 NYY SRBYNNAYNPNNI 12 3 0 0 0 0 0 * TOTALS * 12 3 0 0 0 0 0 ***** BOTTOM OF LIST ***** </pre> </div>

Process to analyze a high-cost package SQL query using costs (continued)

Step	Action
13	To view EXPLAIN and cost information for this package, use format 6. To do so, type FORMAT6 on the command line.
14	Press Enter. Result: !DB/EXPLAIN displays the information about this package using format 6.
15	On the Packages format 6 panel, type E in the Select field of the listed package. <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 10px auto;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 3 CMD ==> SCROLL ==> PAGE P A C K A G E S Cnds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL PACKAGE DATE EXPLAIN TIME STATEMENT COSTS ----- E TTXEMP1 1999/08/23 14:29:52 3,571,336.0 1,551,348.0 6,205,394.0 ***** ***** BOTTOM OF LIST ***** </pre> </div>
16	Press Enter. Result: The package is EXPLAINED. EXPLAINing a package of this size may create many pages of EXPLAIN output. A partial display of the EXPLAIN for this package follows Step 17.
17	Analyze the EXPLAIN output. Read the information in the Recommendations and the access path summary information to ascertain the reason for the high cost. Re-evaluate your SQL content to improve the access path strategy used by DB2.

Explain output

The output of the EXPLAIN requested in Step 15 follows.

```

----- DB/EXPLAIN DB2-D42B ----- LINE 1 OF 2,337
Cmd ==>                               SCROLL ==> PAGE
                                E X P L A I N
Cmds: DO (Menu) GLOBAL (Menu)
-----
PACKAGE= DSNHYCRD  COLLID=DSNHYCRD          CONTOKEN=155403E608555E10
PRECOMPILE=1999/02/13 07:38:37.843832
EXPLAINED=1999/12/09 06:53:19  DEGREE=1
-----
+----- DETAILED PACKAGE INFORMATION -----+
| PACKAGE= DSNHYCRD  COLLECTION= DSNHYCRD
| CREATOR= DCYBU2   OWNER       = DCYBU2   BOUND  = 1999/10/14 08:53:15.248464
| FREED? = N       QUALIFIER = DCYBU2   CREATED= 1999/10/14 12:11:05.333017
| PKSIZE = 1464   SYSENTRIES= 0        PRECOMP= 1999/02/13 19:09:54.148139
| AVSIZE = 2545   DEGREE    = 1        CONTOKN= 155403E608555E10
| LIBRARY= PB.DB2.V420.S26.SDSNDBRM
| VERSION=
| VALID= Y        OPERATIVE= Y   VALIDATE= R   RELEASE=   EXPLAIN= N
| COMMA= N       ISOLATION= S   HOSTLANG= B   SQLERR= N   CHARSET= A
| QUOTE= N       DEFERPREPARE= REMOTE = N   DEC31 = N   MIXED  = N
|                DYNAMICRULES= DEFERPRE= C   KEEPDM= N   REOPT  = N
+-----
+----- RECOMMENDATIONS FOR PACKAGE TTXEMP1 -----+
| RECOMMENDATION B00004
| The plan or package this statement belongs to was bound with the
| option VALIDATE(RUN). Generally the option VALIDATE(BIND) is
| preferred.
|
| When the option VALIDATE(RUN) is specified, DB2 does validity
| checking at BIND time and then rechecks any failures at run time.
| The validity checking at run time can cause catalog contention
| and degraded performance. If possible, ensure that all objects
| exist and all privileges are granted prior to binding a plan or
| package and then choose the VALIDATE(BIND) option.
+-----
+----- STATEMENT NUMBER 1053 TEXT -----+
| DECLARE CURSOR1 CURSOR FOR
| SELECT TEXT , SEQNO
| FROM SYSIBM.SYSVIEWS
| WHERE NAME = :INNAME AND CREATOR = :INAUTH          | |
| ORDER BY SEQNO
+-----
+----- ACCESS PATH STRATEGY FOR STATEMENT NUMBER 103 -----+
| Statement Cost: 779.4
| The cost (timmons) is a rough estimate of resources required to
| execute the SQL statement.
|
| Query Block No. 1
|
| Step 1
|
| Access Info:
|
| This is either a composite table or it is the first table accessed.
| This step accesses T0DB00.TTT0EMP.
| Uses index T0DB00.XEMP2.
| Matching on 1 index column.
| Table pages are accessed.
| List Prefetch is being used (RIDs are being sorted.)
| Column functions are to be decided at run time.
|
| Lock Info:
|
| Intent Share. Lock holder has read-only access.
| Concurrent reads and updates are allowed.
+-----

```

EXPLAIN output (continued)

```

+----- RECOMMENDATIONS FOR STATEMENT NUMBER 103 -----+
RECOMMENDATION 000002
  Tablespace TTTDEXP.TTTSEXP2 has a LOCKSIZE of ANY. Lock
  escalation may occur. If concurrency is of utmost importance for
  applications referencing this tablespace then LOCKSIZE(PAGE) may
  be a better choice.

  When LOCKSIZE ANY is specified, the number of locks that any
  program can hold within a table space is limited by the value
  specified for LOCKS PER TABLE(SPACE) when DB2 was installed.
  When the number of page locks reaches the limit, a tablespace
  lock is acquired and all of the page locks are released. When
  LOCKSIZE PAGE is specified, lock escalation will not occur.

RECOMMENDATION 000002
  Tablespace TTTDEXP.EMPPROJA has a LOCKSIZE of ANY. Lock
  escalation may occur. If concurrency is of utmost importance for
  applications referencing this tablespace then LOCKSIZE(PAGE) may
  be a better choice.
  .
  .

RECOMMENDATION 000016
  Index TDD000.XEMP2 is defined on non read-only
  table TDD000.TTT0EMP and has 4 subpages.

  A subpage is the unit of locking for an index.

  If the table TDD000.TTT0EMP has a high rate of
  INSERTS, DELETES or UPDATES of columns that are included in the
  index key then a small value for subpages (SUBPAGES=1) is
  recommended to avoid index subpage and index page splitting
  and to reduce the amount of locking on higher level index pages.

  If the table TDD000.TTT0EMP has a high rate of
  UPDATES of columns that are not included in the
  index key then a larger value for subpages (SUBPAGES>1) is
  recommended to improve concurrency.

  Consider dropping and recreating the index with a value of
  subpages that meets your installation's standards and meets the
  needs of your application.

RECOMMENDATION 000016
  Index TDD000.XEMP1 is defined on non read-only
  table TDD000.TTT0EMP and has 4 subpages.
  .
  .
  .

+-----+
RECOMMENDATION 000020
  Table TDD000.TTT0DEPT has 3 indexes. This table
  is a non read-only table. Your installation has requested that
  this recommendation be displayed when the number of indexes meets
  or exceeds 3.

  A large number of indexes on non read-only tables may degrade
  performance.

  Each index must be maintained when INSERTS, DELETES and UPDATES
  of a indexed column occur. The index locking overhead and the
  probability of deadlocks increases as the number of indexes
  increases. Utilities may have a longer elapsed time.

  Evaluate the need for each of the indexes defined on this table.
  Consider dropping unneeded index(s).

RECOMMENDATION 000021
  Table TDD000.TTT0EMPPROJACT does not have a clustering
  index.

  A clustering index can provide significant performance advantages
  for SQL statements which process a large number of rows. For
  example, SQL statements which order results, group results or
  have predicates which do comparisons other than equal to.

  Consider creating a clustering index to meet your applications
  needs.
+-----+

```

EXPLAIN output (continued)

```

+-----+
RECOMMENDATION 000022
An index is not being used to access table
T0DB00.TTT@EMPPROJACT. Index(s) are defined on this table.

Consider constructing the SQL statement so that an available
index will be utilized.

IX=T0DB00.XEMPPROJACT2
IX=T0DB00.XEMPPROJACT1

RECOMMENDATION 000022
An index is not being used to access table
T0DB00.TTT@PROJ. Index(s) are defined on this table.

Consider constructing the SQL statement so that an available
index will be utilized.

IX=T0DB00.XPROJ2
IX=T0DB00.XPROJ1

EXCEPTION X00003
List Prefetch

+-----+
----- DETAILED OBJECT INFORMATION FOR STATEMENT NUMBER 103 -----
TS= TTTDEXP.TTTSEXP2 PARTITIONS= 4 LOCKRULE= A
NACTIVE= 0 OBID= 18 CLOSERULE= Y CREATOR= T0DB00
NTABLES= 1 DBID= 261 IMPLICIT= N CREATEBY= T0DB00
SPACE= 192 PSID= 19 STATUS= A BPOOL= BP0
PGSIZE= 4 SEGSIZE= 0 ERASERULE= N DSETPASS=
LOCKMAX= 0 STATSTS= 0001/01/01 00:00:00.000000

TBL= T0DB00.TTT@EMP LOCATION=
RECLN= 97 TBCREATR= TYPE= T
CARD= -1 TBNAME= STATUS= X
NPAGES= -1 CREATEBY= T0DB00 CKFLAG=
PCTPAGES= -1 VALPROC= AUDIT=
KEYOBID= 23 EDPROC= CHILD= 0
OBID= 20 CKRID= PARNT= 0
COLCOUNT= 14 RBA1= 00005462 DATA=
KEYCOLUMNS= 1 RBA2= 00005462
PCTROWCOMP= -1 CREATETS= 1999/12/15 08:07:48.909010
CHECKS= ALTERTS= 1999/12/15 08:07:48.909010
CLUSTERTYPE= STATSTS= 0001/01/01 00:00:00.000000

----- USED INDEXES -----
IX= T0DB00.XEMP2 CREATEBY= T0DB00
FULLKEY= -1 NLEVELS= -1 CLUSTRATIO= 0
FIRSTKEY= -1 NLEAF= -1 CLUSTERED= N
SPACE= 48 COLCOUNT= 1 CLUSTERING= N
PGSIZE= 1024 ERASERULE= N BPOOL= BP0
SUBPAG= 1 CLOSERULE= Y DSETPASS=
OBID= 21 UNIQUE= 0 INDEXSPACE= XEMP2
ISOBID= 22 STATSTS= 0001/01/01 00:00:00.000000
BPOOL= BP0

----- UNUSED INDEXES -----
IX= T0DB00.XEMP1 CREATEBY= T0DB00
FULLKEY= -1 NLEVELS= -1 CLUSTRATIO= 0
FIRSTKEY= -1 NLEAF= -1 CLUSTERED= Y
SPACE= 192 COLCOUNT= 1 CLUSTERING= Y
PGSIZE= 1024 ERASERULE= N BPOOL= BP0
SUBPAG= 1 CLOSERULE= Y DSETPASS=
OBID= 23 UNIQUE= P INDEXSPACE= XEMP1
ISOBID= 24 STATSTS= 0001/01/01 00:00:00.000000
BPOOL= BP0
.
.
.
+-----+

```

Print the Results of a Cost Analysis

Overview

In the previous scenario, you sorted the package list to determine the package with the highest maximum cost and EXPLAINED it. In this scenario, you will use the batch utility to print cost information about package costs in three different formats:

- Highest cost packages
- Packages with the highest cost statements
- Packages with the highest average cost per statement

Commands to perform the cost analyses

Use this table to determine the batch commands required to perform your cost analyses.

Task You Want to Perform	Batch Commands to Use
Determine the packages that have the most expensive SQL statements by <i>total cost (sum)</i> of the SQL statements (cost of statement 1 + cost of statement 2 + ... cost of statement n = sum). Print the list of packages in descending order. This produces a package list that goes from most costly to least costly.	CO CPKG DSORT CTCST PRNT END
Determine the packages that have the highest cost SQL statement. Print the list of packages in descending order by the <i>single most costly</i> SQL statement.	CO CPKG DSORT CMCST PRNT END
Determine the packages that have the most expensive SQL statements by <i>average cost</i> (cost of statement 1 + cost of statement 2 + ... cost of statement n/n). Print the list of packages in descending order from most costly to least costly.	CO CPKG DSORT CACST PRNT END

Analyzing cost by DBRM

You can choose to make your cost analyses by DBRM instead of by package. To do so, you can substitute the command **CDBR** in place of the command **CPKG** in the examples in the preceding section. Alternatively, you can perform an analysis for packages and DBRMs in a single job step. This example first prints the package list by average cost in ascending order, followed by the DBRM list by average cost in ascending order.

```
CO
CPKG
SORT CACST
PRNT
CDBR
SORT CACST
PRNT
END
```

Process to print the package costs

Step	Action
1	Include the batch job steps you developed as SYSIN input in your <i>DB2idUTIL</i> job. Your <i>DB2idUTIL</i> job stream looks like this: <pre> //JOB //KTEBATCH PROC DB@LOAD=, : //EXBATCH.SYSIN DD * CO CPKG DSORT CTCST PRNT DSORT CMCST PRNT DSORT CACST PRNT END /*</pre>
2	Submit the batch job for execution.

Execute a “Merge Since” Extract

Overview

A “merge since” extract merges data with an earlier extract from the same range of plans or packages and also identifies and keeps DB2 deleted data. However, this type of extract only extracts data *since* the last extract for the same range of plans or packages. For a more detailed explanation of the extract, see “The !DB/EXPLAIN Extract Program and the Extract ID” on page 333.

In this scenario you will:

- Access an extract
- Change the extract to a merge since extract
- Execute the merge since extract

Process to execute a “merge since” extract

Step	Action
1	<p>Access an extract from the appropriate DB2 subsystem. Following is some JCL that defined the extract you will access. The extract type is defined within the SYSIN DD (1).</p> <pre data-bbox="573 1073 1248 1224"> : //EXTRACT.SYSIN DD * EXTRACT CATALOG EXTRACT_ID D23A COWNER SYSIBM MERGE 1 PLAN < F PACKAGE BETWEEN AS07HDG AND F007P150 //*</pre>
2	<p>Change the extract to a merge since extract. In order to change an extract to a merge since extract, the last line of the extract JCL must be changed as shown in 2.</p> <pre data-bbox="573 1379 1248 1547"> : //EXTRACT.SYSIN DD * EXTRACT CATALOG EXTRACT_ID D23A COWNER SYSIBM MERGE SINCE 2 PLAN < F PACKAGE BETWEEN AS07HDG AND F007P150 //* 00443015</pre>
3	<p>Reexecute the JCL following the procedures established for your installation to complete the merge since extract.</p>

EXPLAIN SQL Statements in a Source Program and Review the Results

Overview

This unit describes the steps you can take during application development to EXPLAIN SQL statements in a source program. In this scenario, you will

- EXPLAIN SQL statements in an assembler source program
- Gather statistics
- Display the resultant EXPLAIN
- Perform a “what-if” analysis on the output

Process to EXPLAIN SQL in a source program

To EXPLAIN the SQL statements in a source program, follow the steps in the table.

Step	Action
1	Within an ISPF session, access the member containing the source program you want to EXPLAIN.
2	Assume the program is an assembler program. Invoke KTEXPL by typing the extract ID (DB2) and the language format (A for assembler) on the command line: <p style="text-align: center;">KTEXPL DB2 A</p> <p>Note: You need to enter the last two parameters (DB2 A) only if this is the first invocation of KTEXPL, or if these parameters differ from the last KTEXPL invocation.</p>

Process to EXPLAIN SQL in a source program (continued)

Step	Action
3	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Statements display.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D42B ----- LINE 1 OF 13 Cmd ==>> SCROLL ==>> CSR S T A T E M E N T S C m d s : D O (M e n u) G L O B A L (M e n u) S e l e c t s : ? (M e n u) ----- SEL STMT S I C NO T S H SQL TEXT ----- PACKAGE=DSNHYCRD COLLID=DSNHYCRD CONTOKEN=155403E6085555E10 PRECOMPILE=1999/02/13 07:38:37.843832 - 1051 N DECLARE CURSOR1 CURSOR FOR - 1056 N DECLARE CURSOR2 CURSOR FOR - 204 N SELECT TYPE , COLCOUNT - 277 N OPEN CURSOR1 - 278 N FETCH CURSOR1 - 360 N FETCH CURSOR1 - 375 N CLOSE CURSOR1 - 420 N OPEN CURSOR2 - 421 N FETCH CURSOR2 - 637 N FETCH CURSOR2 - 652 N CLOSE CURSOR2 ***** BOTTOM OF LIST ***** </pre> </div>
4	<p>You want to view the full SQL for all EXPLAINable statements. On the Statements display:</p> <ul style="list-style-type: none"> ● Enter SHOWE on the command line, or ● Type DO on the command line to obtain the DO Commands panel. On that panel, select the SHOWE Command.
5	<p>Press Enter</p> <p>Result: !DB/EXPLAIN expands the Statements display to show the full SQL for all EXPLAINable statements.</p>

Process to EXPLAIN SQL in a source program (continued)

Step	Action
6	<p>You want to EXPLAIN a specific statement and gather statistics as a prelude to performing up 4at-if analysis. Type E in the SELECT column next to the statement you want to EXPLAIN.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D42B ----- LINE 1 OF 13 Cmd ==>> SCROLL ==>> CSR S T A T E M E N T S C m d s : D O (M e n u) G L O B A L (M e n u) S e l e c t s : ? (M e n u) ----- S T M T S I C S E L N O T S H S Q L T E X T ----- P A C K A G E = D S N H Y C R D C O L L I D = D S N H Y C R D C O N T O K E N = 1 5 5 4 0 3 E 6 0 8 5 5 5 E 1 0 P R E C O M P I L E = 1 9 9 9 / 0 2 / 1 3 0 7 : 3 8 : 3 7 . 8 4 3 8 3 2 - 1 0 5 1 N D E C L A R E C U R S O R 1 C U R S O R F O R - 1 0 5 6 N D E C L A R E C U R S O R 2 C U R S O R F O R - 2 0 4 N S E L E C T T Y P E , C O L C O U N T - 2 7 7 N O P E N C U R S O R 1 - 2 7 8 N F E T C H C U R S O R 1 - 3 6 0 N F E T C H C U R S O R 1 - 3 7 5 N C L O S E C U R S O R 1 - 4 2 0 N O P E N C U R S O R 2 - 4 2 1 N F E T C H C U R S O R 2 - 6 3 7 N F E T C H C U R S O R 2 - 6 5 2 N C L O S E C U R S O R 2 ***** BOTTOM OF LIST ***** </pre> </div>
7	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the EXPLAIN and gathers DB2 catalog statistics (RUNSTATS) which are displayed along with the EXPLAIN output. Gathering RUNSTATS is prerequisite to performing a what-if analysis.</p>
8	<p>Review the EXPLAIN display. It contains the following sections:</p> <ul style="list-style-type: none"> ● SQL text of the statement being EXPLAINED ● Access path information (type of access such as matching index scan or table space scan) and relative cost estimates ● Recommendations to improve the performance of the SQL statement ● DB2 catalog statistics for any accessed objects such as tables or indexes <p>A partial EXPLAIN illustrating the foregoing information follows this table.</p>
9	<p>Press End to return to the Statements panel.</p>

EXPLAIN output

The EXPLAIN for statement 254 follows.

EXPLAIN SQL Statements in a Source Program and Review the Results

```

----- DB/EXPLAIN DB2=D31A ----- LINE1 OF 70
CMD ==>                                SCROLL ==> PAGE
                                E X P L A I N
Cmds: DO (Menu) GLOBAL (Menu)
-----
SOURCE=TDTD00.TE240BAS.ASM(KTEPSYEX)
EXPLAINED=1999/09/19 07:45:39 DEGREE=1
----- STATEMENT NUMBER 254 TEXT -----
SELECT NAME , CREATOR , TBNAME , TBCREATOR
INTO :NAME , :CREATOR , :TBNAME , :TBCREATOR
FROM SYSIBM.SYSSYNONYMS
WHERE SYSIBM.SYSSYNONYMS.NAME = :NAME AND SYSIBM.SYSSYNONYMS.CREATOR =
      :K_CREATOR
----- ACCESS PATH STRATEGY FOR STATEMENT NUMBER 254 -----
Statement Cost:      8.0
The cost (timerons) is a rough estimate of resources required to
execute the SQL statement.

Query Block No. 1

Step 1

Access Info:

This is either a composite table or it is the first table accessed.
This step accesses SYSIBM.SYSSYNONYMS
Uses index SYSIBM.DSNDYX01.
Matching on 2 index columns.
Table pages are accessed.
Column functions are to be decided at run time.

Lock Info:

Intent Share. Lock holder has read-only access.
Concurrent reads and updates are allowed.
----- RECOMMENDATIONS FOR STATEMENT NUMBER 254 -----
NO RECOMMENDATIONS FOR STATEMENT
----- DETAILED OBJECT INFORMATION FOR STATEMENT NUMBER 254 -----
TS= DSND06.SYSDBASE      PARTITIONS= 0  LOCKRULE = A
NACTIVE= 4680           OBID= 1          CLOSERULE= N  CREATOR = SYSIBM
NTABLES= 14            DBID= 6          IMPLICIT = N  CREATEBY= SYSIBM
SPACE = 0              PSID= 9          STATUS = A   BPOOL = BP0
PGSIZE = 4             SEGSIZE= 0       ERASERULE= N  DSETPASS=
LOCKMAX= 0             STATSTS= 0000/00/00 00:00:00.000000

TBL= SYSIBM.SYSSYNONYMS      LOCATION=
RECLEN= 81                  TBCREATR=                TYPE = T
CARD = 276                  TBNAME =                  STATUS=
NPAGES= 163                 CREATEBY= SYSIBM          CKFLAG=
PCTPAGES = 3                VALPROC =                  AUDIT =
KEYOBID = 0                 EDPROC =                  CHILD = 0
OBID = 33                   CKRID =                    PARN1 = 1
COLCOUNT = 6              RBA1 = 00000000          DATAC =
KEYCOLUMNS= 0             RBA2 = 00000000
PCTROWCOMP= 0             CREATETS= 1998/04/01 00:00:00.000000
CHECKS = 0                 ALTRTS = 1998/04/01 00:00:00.000000
CLUSTERTYPE=              STATSTS = 0000/00/00 00:00:00.000000

----- USED INDEXES -----
IX= SYSIBM.DSNDYX01        CREATEBY = SYSIBM
FULLKEY = 276             NLEVELS= 2                CLUSTRATIO= 62
FIRSTKEY= 40              NLEAF= 40                 CLUSTERED = N
SPACE = 0                 COLCOUNT = 2             CLUSTERING= N
PGSIZE= 4096             ERASERULE= N              BPOOL = BP0
SUBPAG= 1                 CLOSERULE= N              DSETPASS =
OBID = 67                 UNIQUE = U                 INDEXSPACE= DSNDYX01
ISOBID= 99                STATSTS = 0000/00/00 00:00:00.000000
BPOOL = BP0
-----
***** BOTTOM OF LIST *****

```

Process to perform a what-if analysis

In a test environment, there are often relatively small amounts of data. This might lead you to the conclusion that a given access path is performing well. In general, execution time for a given statement will increase in proportion to the increased size of the accessed objects. To understand the implications of increased object size, you can run a what-if by following this process.

Step	Action
<p>1</p>	<p>To obtain the Whatif Analysis panel, type the W select in the Selects column on the Statements display next to the statement you want to analyze.</p> <pre> ----- DB/EXPLAIN DB2=D42B ----- LINE 1 OF 13 Cmd ==>> SCROLL ==>> CSR S T A T E M E N T S Cmds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- STMT S I C SEL NO T S H SQL TEXT ----- PACKAGE=DSNHYCRD COLLID=DSNHYCRD CONTOKEN=155403E6085555E10 PRECOMPILE=1999/02/13 07:38:37.843832 - 1051 N DECLARE CURSOR1 CURSOR FOR - 1056 N DECLARE CURSOR2 CURSOR FOR - 204 N SELECT TYPE , COLCOUNT - 277 N OPEN CURSOR1 - 278 N FETCH CURSOR1 - 360 N FETCH CURSOR1 - 375 N CLOSE CURSOR1 - 420 N OPEN CURSOR2 - 421 N FETCH CURSOR2 - 637 N FETCH CURSOR2 - 652 N CLOSE CURSOR2 ***** BOTTOM OF LIST ***** </pre>
<p>2</p>	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Whatif Analysis panel in Format 1, Access Path format.</p> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE1 OF 12 CMD ==>>> SCROLL ==>>> PAGE W H A T I F A N A L Y S I S Cmds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- The fields marked with * can be updated. TS= DSND06.SYSDBASE CREATOR= SYSIBM *NACTIVE= 4680 - TBL= SYSIBM.SYSSYNONYMS LOCATION= *NPAGES= 163 *PCTROWCOMP = 0 *CARD = 276 ----- USED INDEXES ----- - IX= SYSIBM.DSNDYX01 *FULLKEY = 276 *NLEVELS= 2 *CLUSTRATIO= 62 *FIRSTKEY= 40 *NLEAF = 40 ***** BOTTOM OF LIST ***** </pre>

Process to perform a what-if analysis (continued)

Step	Action
3	<p>Examine the fields on this panel. When you execute this scenario in your environment, because you are testing your application, you may observe that there are few if any RUNSTATS gathered (as made clear by the -1 values in many of the fields). Specific fields of interest and their meanings are:</p> <p>CARD Number of rows in the table</p> <p>RECLEN Length of the row</p> <p>FULLKEY Cardinality (number of different values) of the full key of the index.</p> <p>FIRSTKEY Cardinality (number of different values) of the first column of the index.</p> <p>NLEVELS Levels of the index (usually a value from 1 through 4)</p> <p>NLEAF Number of leaf pages in the index.</p> <p>CLUSTRATIO How closely the sequence of the rows in the table matches the sort order in the index definition. Values above 95% usually result in the most desirable access path.</p> <p>NACTIVE Number of pages in the table space.</p> <p>For full information about these fields, see the Panel Reference information for the Whatif Panel. You may also want to reference the <i>IBM DB2 Administration Guide</i>, which contains expanded information about these fields.</p>
4	Update the fields with projected values. Use values that are typical for your installation for your analysis.
5	<p>On the updated Whatif Analysis panel:</p> <ul style="list-style-type: none"> ● Type WIFL on the command line, or ● Type DO on the command line to obtain the DO Commands panel. On that panel, select the WIFL command.

Process to perform a what-if analysis (continued)

Step	Action
6	<p data-bbox="570 342 727 373">Press Enter.</p> <p data-bbox="570 394 1403 527">Result: !DB/EXPLAIN runs the analysis using the simulated RUNSTATS. The DB2 catalog is changed temporarily for this EXPLAIN, but there is no impact on the access path of other applications accessing the same tables.</p> <p data-bbox="570 548 1430 646">Use of the WIFUL Command, which updates the DB2 catalog, is not recommended before a review of its effect on other running applications is performed.</p>

Set Up General Housekeeping Options to Customize !DB/EXPLAIN

Overview

In this scenario you will

- Access the Housekeeping panel
- Change the default data formats
- Change the default panel sorts
- Change the library search order

Begin at the !DB/EXPLAIN Administration Menu.

Process to set up general housekeeping options to customize !DB/EXPLAIN

Step	Action
1	On !DB/EXPLAIN's Administration Menu, type 5 to select Housekeeping.
2	Press Enter. Result: !DB/EXPLAIN displays the Housekeeping Menu panel.
3	Type 18 in the Option field to display the Data Formats panel. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==> DB/EXPLAIN HOUSEKEEPING MENU Option ==> 0 Global Data Set Information 14 Extract Processing Defaults 1 Global Configuration Information 15 Tuning Parameters 2 Global Print Options 16 Build Tuning Parameters 3 DB2 Subsystem Information 17 Explain Display Tuning Parameters 4 DB/EXPLAIN Configuration 18 Data Formats 5 Library Search Order 19 Panel Formats 6 Selection Masking 20 Panel Sorts 7 SET Defaults 21 Panel Filters 8 Output Options 22 SQL Parameters for KTEXPL 9 Miscellaneous Defaults 23 Exception Options 10 Miscellaneous Defaults 24 Plan BIND Overrides 11 Explain / SQL Defaults 25 Package BIND Overrides 12 SQL Defaults 26 BIND Compare Options 13 Compare Options 27 Recommendations Press ENTER to process. Press END to cancel. </pre> </div>

Process to set up general housekeeping options to customize !DB/EXPLAIN (continued)

Step	Action
4	<p>On the Data Formats panel, change the default formats.</p> <ul style="list-style-type: none"> Type 4 in the Date and Timestamp fields and 3 in the Count and Costs1 fields to change those formats throughout !DB/EXPLAIN. Type Y in the Permanent field to make the changes permanent. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> D A T A F O R M A T S PERMANENT? ==> Y (Y Yes N No) TIME ==> 1 1 HH:MM:SS 2 HH.MM.SS DATE ==> 4 1 YYYY/MM/DD 2 YYYY-MM-DD 3 MM/DD/YYYY 4 MM-DD-YYYY 5 DD.MM.YYYY 6 DD/MM/YYYY 7 YY/MM/DD 8 YY-MM-DD 9 MM/DD/YY 10 MM-DD-YY 11 DD.MM.YY 12 DD/MM/YY TIMESTAMP ==> 4 1 YYYY/MM/DD HH:MM:SS.TTTTTT 2 YYYY-MM-DD HH:MM:SS.TTTTTT 3 MM/DD/YYYY HH:MM:SS.TTTTTT 4 MM-DD-YYYY HH:MM:SS.TTTTTT 5 DD.MM.YYYY HH:MM:SS.TTTTTT 6 DD/MM/YYYY HH:MM:SS.TTTTTT 7 YYYY/MM/DD HH:MM:SS 8 YYYY-MM-DD HH:MM:SS 9 MM/DD/YYYY HH:MM:SS 10 MM-DD-YYYY HH:MM:SS 11 DD.MM.YYYY HH:MM:SS 12 DD/MM/YYYY HH:MM:SS COUNT ==> 3 1 9999999 2 9.999.999 3 9,999,999 COSTS1 ==> 3 1 9999999 2 9.999.999,99 3 9,999,999.99 COSTS2 ==> 2 1 /1000 2 ASIS 3 /100 4 /10000 5 /100000 6 /4096 ENTER to process END to cancel </pre> </div>

Process to set up general housekeeping options to customize !DB/EXPLAIN (continued)

Step	Action
5	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Housekeeping Menu with a message indicating the update.</p> <pre> ----- DB/EXPLAIN DB2=D42B Permanent changes made Cmd ==> 18 DB/EXPLAIN HOUSEKEEPING MENU Option ==> 0 Global Data Set Information 14 Extract Processing Defaults 1 Global Configuration Information 15 Tuning Parameters 2 Global Print Options 16 Build Tuning Parameters 3 DB2 Subsystem Information 17 Explain Display Tuning Parameters 4 DB/EXPLAIN Configuration 18 Data Formats 5 Library Search Order 19 Panel Formats 6 Selection Masking 20 Panel Sorts 7 SET Defaults 21 Panel Filters 8 Output Options 22 SQL Parameters for KTEXPL 9 Miscellaneous Defaults 23 Exception Options 10 Miscellaneous Defaults 24 Plan BIND Overrides 11 Explain / SQL Defaults 25 Package BIND Overrides 12 SQL Defaults 26 BIND Compare Options 13 Compare Options 27 Recommendations Press ENTER to process. Press END to cancel. </pre>
6	<p>On the Housekeeping Menu, type 18 in the Select Function field to display the Panel Sorts panel.</p>
7	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Panel Sorts panel.</p> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> PANEL SORTS PERMANENT? ==> N (Y Yes N No) COLLECTIONS QCOL A COSTS CNAME A DBRMS DBRM D HOST VARIABLES HOST A LIBRARIES LLIB A PACKAGES KPACK A PLANS PLAN A PACKAGE LISTS PLOC A PLAN CONNECTIONS PSYS A PACKAGE CONNECTIONS KSYS A TABLES TTBL A ENTER to process END to cancel </pre>

Process to set up general housekeeping options to customize !DB/EXPLAIN (continued)

Step	Action
8	<p>On the Panel Sorts panel, type DBRM in the Sort Command field for DBRMs so the DBRM panel will sort its display list by DBRM name.</p> <ul style="list-style-type: none"> ● Type D in the Sort Type field for DBRMs so the DBRM panel will sort the list in descending order. ● Type N in the Permanent field to specify temporary changes.
9	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Housekeeping Menu showing the temporary update (for this session only).</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D42B Temporary changes made Cmd ==> DB/EXPLAIN HOUSEKEEPING MENU Option ==> 0 Global Data Set Information 14 Extract Processing Defaults 1 Global Configuration Information 15 Tuning Parameters 2 Global Print Options 16 Build Tuning Parameters 3 DB2 Subsystem Information 17 Explain Display Tuning Parameters 4 DB/EXPLAIN Configuration 18 Data Formats 5 Library Search Order 19 Panel Formats 6 Selection Masking 20 Panel Sorts 7 SET Defaults 21 Panel Filters 8 Output Options 22 SQL Parameters for KTEXPL 9 Miscellaneous Defaults 23 Exception Options 10 Miscellaneous Defaults 24 Plan BIND Overrides 11 Explain / SQL Defaults 25 Package BIND Overrides 12 SQL Defaults 26 BIND Compare Options 13 Compare Options 27 Recommendations Press ENTER to process. Press END to cancel. </pre> </div>
10	<p>You now want to change the library search order. On the Housekeeping Menu, type 5 in the Select Function to display the Library Search Order panel.</p>

Process to set up general housekeeping options to customize !DB/EXPLAIN (continued)

Step	Action
11	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Library Search Order panel. To change the library search order,;</p> <ul style="list-style-type: none"> ● Type F in the Search Order field to specify that the library from the DB2 Catalog should be searched first. ● Type "DEV" SUBSTR(1,4,&DBRM) in the High Level Qualifier #1 field. ● Type &SSID in the Mid Level Qualifier #1 field. ● Type "DBRMLIB" in the Low Level Qualifier #1 field. ● Type N in the Permanent field to specify temporary changes. <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> LIBRARY SEARCH ORDER PERMANENT? ==> N (Y Yes N No) SEARCH ORDER ==> F (Search library specified in DB2 catalog. 0=Only N=Never F=First L=Last) HIGH LEVEL QUALIFIER#1 ==> "DEV" SUBSTR(1,4,&DBRM) MID LEVEL QUALIFIER#1 ==> &SSID LOW LEVEL QUALIFIER#1 ==> "DBRMLIB" HIGH LEVEL QUALIFIER#2 ==> MID LEVEL QUALIFIER#2 ==> LOW LEVEL QUALIFIER#2 ==> HIGH LEVEL QUALIFIER#3 ==> MID LEVEL QUALIFIER#3 ==> LOW LEVEL QUALIFIER#3 ==> HIGH LEVEL QUALIFIER#4 ==> MID LEVEL QUALIFIER#4 ==> LOW LEVEL QUALIFIER#4 ==> ENTER to process END to cancel </pre> </div>

Process to set up general housekeeping options to customize !DB/EXPLAIN (continued)

Step	Action
12	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Housekeeping Menu panel.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D42B Temporary changes made Cmd ==>> DB/EXPLAIN HOUSEKEEPING MENU Option ==>> 0 Global Data Set Information 14 Extract Processing Defaults 1 Global Configuration Information 15 Tuning Parameters 2 Global Print Options 16 Build Tuning Parameters 3 DB2 Subsystem Information 17 Explain Display Tuning Parameters 4 DB/EXPLAIN Configuration 18 Data Formats 5 Library Search Order 19 Panel Formats 6 Selection Masking 20 Panel Sorts 7 SET Defaults 21 Panel Filters 8 Output Options 22 SQL Parameters for KTEXPL 9 Miscellaneous Defaults 23 Exception Options 10 Miscellaneous Defaults 24 Plan BIND Overrides 11 Explain / SQL Defaults 25 Package BIND Overrides 12 SQL Defaults 26 BIND Compare Options 13 Compare Options 27 Recommendations Press ENTER to process. Press END to cancel. </pre> </div> <p>Assuming the SSID = D22A and the DBRM = OILS0001, the new library search order (for this session only) is</p> <ol style="list-style-type: none"> 1. The library from the DB2 Catalog 2. DEVOILS.D22A.DBRMLIB

Convert a Plan DBRM to a Package Using the Z Select

Overview

In this scenario you will

- View a listing of all DBRMs
- Select a plan DBRM to convert to a package
- Specify the BIND parameters of the conversion
- Execute the conversion

All associated plans point to the resulting package.

Process to convert a plan DBRM to a package using the Z select

Begin at the !DB/EXPLAIN Primary Menu. Remember that your default formats may be different than the ones shown.

Step	Action
1	On !DB/EXPLAIN's Primary Menu, type 2 to view a listing of all DBRMs.
2	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the DBRMS panel.</p> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE1 OF 172 CMD ==> SCROLL ==> PAGE D B R M S Cmds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL DBRM VQCCMDH LUOHXEL PLAN STMTS SELECTS DELETES UPDATES INSERTS LCK LCK ----- - ALTSQLMU NNNAN B DBT925R 17 0 0 7 0 0 0 - ALTSQLUC NNNAN B DBT925R 14 0 0 1 2 0 0 - ALTSQLUM NNNAN B DBT925R 3 0 0 0 1 0 0 </pre>

Process to convert a plan DBRM to a package using the Z select (continued)

Step	Action
3	<p>On the DBRMS panel, select a plan DBRM to convert to a package by typing Z in the SEL column of the DBRM you want to convert.</p> <div data-bbox="573 474 1248 737" style="border: 1px solid black; padding: 5px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> SCROLL ==> PAGE D B R M S Cmds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL DBRM VQCCMDH LUOHXEL PLAN STMTS SELECTS DELETES UPDATES INSERTS LCK LCK ----- Z ALTSQMU NNNAN B DBT925R 17 0 0 7 0 0 0 - ALTSQUC NNNAN B DBT925R 14 0 0 1 2 0 0 _ ALTSQUM NNNAN B DBT925R 3 0 0 0 1 0 0 </pre> </div>
4	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Package Bind panel. On this panel, you can specify the BIND parameters to be in effect for the conversion.</p>
5	<p>On the Package Bind panel:</p> <ul style="list-style-type: none"> ● Select Add by typing A in the Action field. ● Type the name of the Collection in the Collection field. ● Type Y or N in the Currentdata field based on whether or not data currency is required for all cursors when performing the conversion. <p>You can also modify other default values to suit your requirements.</p> <div data-bbox="573 1266 1248 1745" style="border: 1px solid black; padding: 5px;"> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==> P A C K A G E B I N D BIND or REBIND : BIND Action : R (R Replace A Add) Flag : I (I / W / E / C) Package : KTEEXDB3 Isolation: S (R RR T RS CS U UR) Location : Release : C (C Commit D Deallocate) Collection : CANDLE_SYSTEM Validate : B (R Run B Bind) Owner : MYUSER Explain : N (Y Yes N No) Qualifier : SYSIBM Sqlerror : N (N Nopackage C Continue) Dynamicrules: (B / R / blank) Degree : 1 (1 ANY) Keepdynamic : N (Y Yes N No) Deferprep: (Y Yes N No blank) Reopt(VARS) : N (Y Yes N No) Currentdata: Y (Y Yes N No) Version : V500 Enable/Disable Connections?: * (Y Yes N No * ALL) Remote ENABLE? : (Y Yes N No) Source: Library : CANDLE.DBTOOLS.DBRM DBRM: KTEEXDB3 - or - Collection : Package : Version : ENTER to process END to cancel </pre> </div>

Process to convert a plan DBRM to a package using the Z select (continued)

Step	Action
6	<p>Press Enter. If you have previously specified Y in the Display Override Panel for Convert field on the Plan BIND Overrides panel, !DB/EXPLAIN now displays the Plan BIND Overrides panel. Type N in that field.</p> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==> P A C K A G E B I N D O V E R R I D E S PERMANENT? ==> N (Y Yes N No) Use Overrides for Command? Y Select? N Convert? O Compare? N Display Override Panel for Command? Y Select? N Convert? N Compare? N Use Library Search Order for Command? Y Select? P Convert? P Compare? P Location : Action : (R Replace A Add) Collection : Flag : (I / W / E / C) Owner : Isolation : (R RR T RS S CS U UR) Qualifier : Release : (C Commit D Deallocate) Remote ENABLE : Y (Y Yes N No) Validate : (R Run B Bind) BIND COPY : (Y Yes N No) Explain : (Y Yes N No) Deferprep : (Y Yes N No) Sqlerror : (N Nopackage C Continue) Keepdynamic : (Y Yes N No) Degree : (1 / ANY) Reopt(VARS) : (Y Yes N No) Currentdata : (Y Yes N No) Collection : Dynamicrules : (B / R / blank) Specification for Convert: &DBRM Press ENTER to process. Press END to cancel. </pre>
7	<p>Press Enter. If you have previously specified Y in the Display Override Panel for Convert field on the Package Bind Overrides panel, !DB/EXPLAIN displays the Package BIND Overrides panel. Type N in that field.</p> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==> P L A N B I N D O V E R R I D E S PERMANENT? ==> N (Y Yes N No) Use Overrides for Command? Y Select? N Convert? O Compare? N Display Override Panel for Command? Y Select? N Convert? N Compare? N Use Library Search Order for Command? Y Select? Y Convert? Y Compare? Y Owner : Action : (R Replace A Add) Qualifier : Retain : (Y Yes N No) Cache Size : Flag : (I / W / E / C) Current Server: Isolation : (R RR R RS S CS U UR) Current Data : (Yes N No) Acquire : (U Use A Allocate) SQLRULES : (D DB2 S STD) Release : (C Commit D Deallocate) Keepdynamic : (Y Yes N No) Validate : (R Run B Bind) REOPT(VARS) : (Y Yes N No) Explain : (Y Yes N No) Dynamicrules : (B / blank) Deferprep : (Y Yes N No) Gen PKLIST : (N No Y Yes) Degree : (1 / ANY) C Convert Only) Disconnect: (E Explicit A Automatic C Conditional) Collection Specification: Press ENTER to process. Press END to cancel. </pre>

Process to convert a plan DBRM to a package using the Z select (continued)

Step	Action
8	<p>Press Enter.</p> <p>Result: If you so specified on the Output Options Defaults panel, !DB/EXPLAIN displays the Output Options panel. On the Output Options panel, you can either press Enter to accept the default member or type in a new member.</p> <div data-bbox="571 558 1250 1024" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> O U T P U T O P T I O N S You have requested a function which will output data to a PDS member. Specify the PDS and member name. Library: TDDB.DBTOOLS.E230.USER Member: BIND ENTER to accept values END to cancel request </pre> </div>
9	<p>Press Enter to accept the default.</p> <p>Result: !DB/EXPLAIN displays the Process Member panel. Type 1 in the Select Action field to execute the conversion.</p> <div data-bbox="571 1197 1250 1558" style="border: 1px solid black; padding: 10px;"> <pre> PROCESS MEMBER ----- DB/EXPLAIN DB2=D31A ----- BIND cmds saved Cmd ==> 99/09/17 14:55 Specify next action to be performed. Select Action ==> 1 1. UPDATE NOW Invoke DB2 to execute the member. 2. EDIT Edit the member. 3. DELETE Delete the member (and exit). 4. PRINT Print the member. 5. DO LATER Changes are saved, exit now. </pre> </div>
10	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN converts the selected plan DBRM to a package.</p>

Convert Multiple Plan DBRMs to Packages Using the BINDCNV Command

Overview

In this scenario you will

- View a listing of all DBRMs
- Filter DBRMs
- Issue the BINDCNV command to convert the DBRMs that comprise a given plan to packages and rebind the plan.
- Execute the output

Process to convert multiple plan DBRMs to packages using BINDCNV

Follow these steps to convert multiple plan DBRMs to packages in a single step using the BINDCNV command.

Step	Action
1	From the !DB/EXPLAIN Primary Menu, type 2 to view a listing of all DBRMs.
2	Press Enter. Result: !DB/EXPLAIN displays the DBRMS panel.
3	Filter the DBRMs for all DBRMs with more than one plan. On the command line of the DBRMS panel, type FILT? .

```

----- DB/EXPLAIN DB2=031A ----- LINE1 OF 172
CMD ==>  FILT?                               SCROLL ==> PAGE

                                D B R M S
Cmnds: DO (Menu) GLOBAL (Menu)                Selects: ? (Menu)
-----
SEL DBRM      VQCCMDH          LUOHXEL PLAN      STMTS  SELECTS  DELETES  UPDATES  INSERTS  LCK  LCK
-----
-  ALTSQMJU  NNNAN  B  DBT925R      17      0      0      7      0  0  0
-  ALTSQJUC  NNNAN  B  DBT925R      14      0      0      1      2  0  0
-  ALTSQJUM  NNNAN  B  DBT925R       3      0      0      0      1  0  0

```

Process to convert multiple plan DBRMs to packages using BINDCNV (continued)

Step	Action
4	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the DBRM Filtering panel. On that panel:</p> <ul style="list-style-type: none"> ● change the FILTER value to A ● set the operand of D#PL to > ● set the value of D#PL to 1 <pre> ----- DB/EXPLAIN DB2=D42B ----- LINE1 OF 39 D B R M F I L T E R I N G CMD5: GLOBAL (MENU) ----- FILTER ==>> A (A AND R OR O OFF) OP VALUE DBRM DBRM NAME DPLN PLAN NAME DLIB DVER LIBRARY VERS OP VALUE OP VALUE DMIXD MIXED (Y / N) DQUOT QUOTE (Y / N) DDC31 DEC31 (Y / -) DCHRS CHARSET (K / A) DCOMM COMMA (Y / N) DHLNG HOSTLANG (B/C/D/F/P/2/3/4) DMISS STMT MISSING? (Y / N) B ASM C COBOL D C DVRF VERIFIED? (Y/N/E/T) F FORTRAN P PL/I 2 COBOL2 3 00-COBOL 4 C++ OP VALUE DPCDT (PRECOMPILE TIME - YYYYMMDDHHMMSS) DEXDT (EXPLAIN TIME - YYYYMMDDHHMMSS) DMCST (MAXIMUM STATEMENT COST) DACST (AVERAGE STATEMENT COST) DTCST (TOTAL STATEMENT COST) D#PL > 1 (NUMBER OF PLANS) D#TL (TOTAL NUMBER OF STATEMENTS) ----- S Q L C O U N T S D B R M S T M T S ----- OP VALUE OP VALUE OP VALUE D#AL (ALTER) D#EC (EXECUTE) D#RL (ROLLBAC) D#CA (CALL) D#EX (EXPLAIN) D#SE (SELECT) D#CL (CLOSE) D#FT (FETCH) D#ST (SET) D#CM (COMMIT) D#GR (GRANT) D#SC (SET CON) D#CN (CONNECT) D#IN (INSERT) D#SD (SET DEG) D#CR (CREATE) D#LS (LOCK SHR) D#SH (SET HOS) D#DC (DCL CRS) D#LX (LOCK XCL) D#SP (SET PKG) D#DM (DCL STM) D#OP (OPEN) D#SR (SET RUL) ***** BOTTOM OF LIST ***** </pre>
5	Press End to display the filtered DBRMS panel.

Process to convert multiple plan DBRMs to packages using BINDCNV (continued)

Step	Action
6	<p>On the filtered DBRMs panel, issue the BINDCNV command by typing BINDCNV on the command line.</p> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE 10F 23(F) CMD ==> BINDCNV SCROLL ==> CSR C mds: DO (Menu) GLOBAL (Menu) D B R M S Selects: ? (Menu) ----- SEL DBRM VQCCMDH STMTS SELECTS DELETES UPDATES INSERTS LCK LCK LUOHXEL PLAN ----- - DSDDB2UP NNNAN B DBT925E 9 1 0 0 1 0 1 - EXPCB02 NNNAN C EXPPLN10 25 0 0 1 0 0 0 - EXPCB04 NNNAN C EXPPLN4 25 0 0 1 0 0 0 </pre>
7	<p>Press Enter.</p> <p>If you have previously specified Y in the Display Override Panel for Convert field on the Plan BIND Overrides panel, !DB/EXPLAIN now displays the Plan BIND Overrides panel. Type N in that field.</p> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==> P L A N B I N D O V E R R I D E S PERMANENT? ==> N (Y Yes N No) Use Overrides for Command? Y Select? N Convert? 0 Compare? N Display Override Panel for Command? Y Select? N Convert? N Compare? N Use Library Search Order for Command? Y Select? Y Convert? Y Compare? Y Owner : Action : (R Replace A Add) Qualifier : Retain : (Y Yes N No) : Flag : (I / W / E / C) Cache Size : Isolation : (R RR R RS S CS U UR) Current Server: Acquire : (U Use A Allocate) : Release : (C Commit D Deallocate) Current Data : (Yes N No) SQLRULES : (D DB2 S STD) Validate : (R Run B Bind) Keepdynamic : (Y Yes N No) Explain : (Y Yes N No) REOPT(VARS) : (Y Yes N No) Defereprep : (Y Yes N No) Dynamicrules : (B / blank) Degree : (1 / ANY) Gen PKLIST : (N No Y Yes) Disconnect: (E Explicit A Automatic : C Convert Only) C Conditional) Collection Specification: Press ENTER to process. Press END to cancel. </pre>

Process to convert multiple plan DBRMs to packages using BINDCNV (continued)

Step	Action
<p>8</p>	<p>Press Enter.</p> <p>If you have previously specified Y in the Display Override Panel for Convert field on the Package Bind Overrides panel, !DB/EXPLAIN displays the Package BIND Overrides panel. Type N in that field.</p> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==>> P A C K A G E B I N D O V E R R I D E S PERMANENT? ==>> N (Y Yes N No) Use Overrides for Command? Y Select? N Convert? O Compare? N Display Override Panel for Command? Y Select? N Convert? N Compare? N Use Library Search Order for Command? Y Select? P Convert? P Compare? P Location : Action : (R Replace A Add) Collection : Flag : (I / W / E / C) Owner : Isolation : (R RR T RS S CS U UR) Qualifier : Release : (C Commit D Deallocate) Remote ENABLE : Y (Y Yes N No) Validate : (R Run B Bind) BIND COPY : (Y Yes N No) Explain : (Y Yes N No) Deferprep : (Y Yes N No) Sqlerror : (N Nopackage C Continue) Keepdynamic : (Y Yes N No) Degree : (1 / ANY) Reopt(VARS) : (Y Yes N No) Currentdata : (Y Yes N No) Collection : Dynamicrules: (B / R / blank) Specification for Convert: &DBRM Press ENTER to process. Press END to cancel. </pre>
<p>9</p>	<p>Press Enter.</p> <p>Result: If you so specified on the Output Options Defaults panel, !DB/EXPLAIN displays the Output Options panel to permit you to either Press Enter to accept the default options or modify the output options by typing over them.</p> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==>> O U T P U T O P T I O N S You have requested a function which will output data to a PDS member. Specify the PDS and member name. Library: TDDB.DE2.V230.USER Member: BIND ENTER to accept values END to cancel request </pre>

Process to convert multiple plan DBRMs to packages using BINDCNV (continued)

Step	Action
10	<p>Press Enter to accept the displayed options.</p> <p>Result: !DB/EXPLAIN displays the Process Member panel. Type 1 in the Select Action field.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> PROCESS MEMBER ----- DB/EXPLAIN DB2=D31A ----- BIND cmds saved Cmd ==> 99/09/25 15:22 Specify next action to be performed. Select Action ==> 1 1. UPDATE NOW Invoke DB2 to execute the member. 2. EDIT Edit the member. 3. DELETE Delete the member (and exit). 4. PRINT Print the member. 5. DO LATER Changes are saved, exit now. </pre> </div>
11	Press Enter to execute the bind convert process.

Convert DBRMs in Multiple Plans to Packages Using the BINDCNV Command

Overview

Prior to V235 of !DB/EXPLAIN, if you wanted to convert DBRMs within a given plan to packages, you first displayed the DBRM object list, then filtered that list by plan name to display all DBRMs associated with a given plan. You could then convert the DBRMs in that plan to packages using the **Z** select for each individual DBRM you wished to convert or the **BINDCNV** command to convert the entire DBRM object list.

To convert DBRMs in multiple plans to packages, you access the DBRMs in each plan separately and proceed as above. For more information, see “Convert Multiple Plan DBRMs to Packages Using the BINDCNV Command” on page 285.

Providing the **BINDCNV** command for plans allows you, in a single step, to convert to packages all DBRMs in a single plan or a set of plans and REBIND those plans.

In this scenario you will

- View a listing of all plans
- Filter the plans to find all plans in an application
- Issue the BINDCNV command to convert DBRMs to packages and perform required BIND or REBIND
- Execute the output

Process to convert DBRMs in multiple plans to packages using BINDCNV

Step	Action
1	On the !DB/EXPLAIN Primary Menu, type 1 to select plans.
2	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the Plans panel.</p> <div data-bbox="571 506 1248 1104" style="border: 1px solid black; padding: 10px;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- LINE 1 OF 544 CMD ==> SCROLL ==> PAGE P L A N S Cmts: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL PLAN COUNT STMTS SELECTS DCL CSR DELETES UPDATES INSERTS XCL LOCK LOCK ----- - ALECV31A 1 31 0 1 0 0 0 0 0 0 0 - A705DPLR 3 21 2 1 0 0 0 0 1 0 1 - A705DPLX 2 18 1 1 0 0 0 0 1 0 1 - CANPLAN 23 240 22 17 3 29 12 0 5 - CANPLNX 3 31 1 2 0 0 0 4 0 5 - CANRWKBC 3 31 3 2 0 0 0 3 0 4 - CANXWKBC 2 22 0 2 0 0 0 3 0 4 - CHES110 2 16 0 1 0 0 0 4 0 0 - CHEX110 2 99 0 14 0 0 0 0 0 0 - CHGPLAN 3 31 3 2 0 0 0 3 0 4 - CHGPLANX 2 22 0 2 0 0 0 3 0 4 - C1PLAN 1 33 0 0 0 0 0 23 0 0 - DBSMUPLN 1 10 0 1 0 0 0 0 0 0 - DEDPLNR 3 21 2 1 0 0 0 1 0 1 - DEDPLNX 2 18 1 1 0 0 0 1 0 1 - DEEPLNQ 2 16 0 1 0 4 0 0 0 0 - DEEPLNR 12 156 4 18 0 4 0 0 0 0 - DEEPLNX 2 99 0 14 0 0 0 0 0 0 - DHD230R 3 21 2 1 0 0 0 1 0 1 - DHD230X 2 18 1 1 0 0 0 1 0 1 - DHE235Q 2 16 0 1 0 4 0 0 0 0 - DHE235R 12 156 4 18 0 4 0 0 0 0 - DHE235X 2 99 0 14 0 0 0 0 0 0 - DHS230R 7 59 4 6 0 4 0 0 0 0 </pre> </div>
3	For purposes of the example, assume you wish to filter the Plans object list for all plans belonging to a given application. On the command line of the plans panel, type FILTA .
4	<p>Press Enter. Then type PLAN=DE*, where DE* identifies the application you are interested in.</p> <p>Note: Steps 3 and 4 illustrate a “fast path” for filtering. You can choose to use the Filtering panel to specify your filters as an alternate method of performing filtering.</p>
5	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays a listing of filtered plans.</p>

Process to convert DBRMs in multiple plans to packages using BINDCNV (continued)

Step	Action
6	<p>Issue the BINDCNV command by typing BINDCNV on the command line.</p> <pre> ----- DB/EXPLAIN DB2=D31A ----- Filter added CMD ==> BINDCNV SCROLL == => PAGE Cnds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- ALL SEL PLAN COUNT STMTS SELECTS DCL CSR DELETES UPDATES INSERTS XCL SHR ----- - DEDPLNR 3 21 2 1 0 0 0 1 0 1 - DEDPLNX 2 18 1 1 0 0 0 1 0 1 - DEEPLNQ 2 16 0 1 0 4 0 0 0 0 - DEEPLNR 12 156 4 18 0 4 0 0 0 0 - DEEPLNX 2 99 0 14 0 0 0 0 0 0 - * TOTALS * 310 7 35 0 8 2 0 2 ***** BOTTOM OF LIST ***** </pre>
7	<p>Press Enter.</p> <p>If you have previously specified Y in the Display Override Panel for Convert field on the Plan BIND Overrides panel, !DB/EXPLAIN now displays the Plan BIND Overrides panel.</p> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==> PLAN BIND OVERRIDES PERMANENT? ==> N (Y Yes N No) Use Overrides for Command? Y Select? N Convert? O Compare? N Display Override Panel for Command? Y Select? N Convert? N Compare? N Use Library Search Order for Command? Y Select? Y Convert? Y Compare? Y Owner : Action : (R Replace A Add) Qualifier : Retain : (Y Yes N No) : Flag : (I / W / E / C) Cache Size : Isolation : (R RR R RS S CS U UR) Current Server: Acquire : (U Use A Allocate) Current Data : (Yes N No) Release : (C Commit D Deallocate) SQLRULES : (D DB2 S STD) Validate : (R Run B Bind) Keepdynamic : (Y Yes N No) Explain : (Y Yes N No) REOPT(VARS) : (Y Yes N No) Deferprep : (Y Yes N No) Dynamicrules : (B / blank) Degree : (I / ANY) Gen PKLIST : (N No Y Yes Disconnect: (E Explicit A Automatic C Convert Only) C Conditional) Collection Specification: Press ENTER to process. Press END to cancel. </pre>

Process to convert DBRMs in multiple plans to packages using BINDCNV (continued)

Step	Action
<p>8</p>	<p>Press Enter.</p> <p>If you have previously specified Y in the Display Override Panel for Convert field on the Package Bind Overrides panel, !DB/EXPLAIN displays the Package BIND Overrides panel.</p> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==> PACKAGE BIND OVERRIDES PERMANENT? ==> N (Y Yes N No) Use Overrides for Command? Y Select? N Convert? O Compare? N Display Override Panel for Command? Y Select? N Convert? N Compare? N Use Library Search Order for Command? Y Select? P Convert? P Compare? P Location : Action : (R Replace A Add) Collection : Flag : (I / W / E / C) Owner : Isolation : (R RR T RS S CS U UR) Qualifier : Release : (C Commit D Deallocate) Remote ENABLE : Y (Y Yes N No) Validate : (R Run B Bind) BIND COPY : (Y Yes N No) Explain : (Y Yes N No) Deferprep : (Y Yes N No) Sqlerror : (N Nopackage C Continue) Keepdynamic : (Y Yes N No) Degree : (1 / ANY) Reopt(VARS) : (Y Yes N No) Currentdata : (Y Yes N No) Collection : Dynamicrules : (B / R / blank) Specification for Convert: &DBRM Press ENTER to process. Press END to cancel. </pre>
<p>9</p>	<p>Press Enter.</p> <p>Result: If you so specified on the Output Options Defaults panel, !DB/EXPLAIN displays the Output Options panel. You can either press Enter to accept the displayed options or type over the displayed information to revise it.</p> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> OUTPUT OPTIONS You have requested a function which will output data to a PDS member. Specify the PDS and member name. Library: TDDB.DE2.V230.USER Member: BIND ENTER to accept values END to cancel request </pre>

Process to convert DBRMs in multiple plans to packages using BINDCNV (continued)

Step	Action
10	Press Enter to accept the displayed options. Result: !DB/EXPLAIN displays the Process Member panel.
11	Type 1 in the Select Action field. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> PROCESS MEMBER ----- DB/EXPLAIN DB2=D31A ----- BIND cmds saved Cmd ==>> 99/09/25 15:22 Specify next action to be performed. Select Action ==>> 1 1. UPDATE NOW Invoke DB2 to execute the member. 2. EDIT Edit the member. 3. DELETE Delete the member (and exit). 4. PRINT Print the member. 5. DO LATER Changes are saved, exit now. </pre> </div>
12	Press Enter to execute the bind convert process.

Change All Plans with an Isolation Level of RR (Repeatable Read) to CS (Cursor Stability)

Overview

In this scenario you will

- View a listing of all plans
- Filter plans
- Issue the BINDREP command
- Execute the output

Process to change all plans with isolation level RR to CS

Step	Action
1	View a listing of all plans by typing 1 on the command line of the !DB/EXPLAIN Primary Menu.
2	Press Enter. Result: !DB/EXPLAIN displays the Plans panel.
3	To filter for all plans with an isolation level of RR (repeatable read), type FILT? on the command line of the Plans panel. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D31A ----- CMD ==> FILT? SCROLL ==> PAGE P L A N S Cnds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL PLAN COUNT STMTS SELECTS DCL CSR ALL DELETES UPDATES INSERTS LOCK LOCK ----- - ALECV31A 1 31 0 1 0 0 0 0 0 0 0 - CANPLAN 23 240 22 17 3 29 12 0 5 - CANPLNX 3 31 1 2 0 0 4 0 5 - CHGPLAN 3 31 3 2 0 0 3 0 4 - CHGPLANX 2 22 0 2 0 0 3 0 4 - DSDPLAN 3 21 2 1 0 0 1 0 1 - DSDPLANX 2 18 1 1 0 0 1 0 1 - DSDPLANZ 2 18 1 1 0 0 1 0 1 - DSDZPLAN 3 21 2 1 0 0 1 0 1 - DSNEDCL 1 3 0 0 0 0 0 0 0 - DSNESPCS 1 17 0 1 0 0 0 0 0 - DSNHYCRD 1 11 1 2 0 0 0 0 0 - DSNTEP2 1 31 0 1 0 0 0 0 0 - DSNTEP31 1 31 0 1 0 0 0 0 0 - DSNTEIAD 1 4 0 0 0 0 0 0 0 - DSNTEIADL 1 20 0 1 0 0 0 0 0 - DSNTEIA31 1 4 0 0 0 0 0 0 0 - DSNTEIB31 1 20 0 1 0 0 0 0 0 - DSN8BH31 1 25 0 3 0 1 0 0 0 - DSN8BP31 1 25 0 3 0 1 0 0 0 - DSN8CC0 3 88 22 8 3 3 3 0 0 </pre> </div>
4	Press Enter. Result: !DB/EXPLAIN displays the Plan Filtering panel.

Process to change all plans with isolation level RR to CS (continued)

Step	Action
5	<p>On the Plan Filtering panel (a portion of which is shown here):</p> <ul style="list-style-type: none"> ● Change the FILTER value to A ● Set the operand of PISO to = ● Set the value of PISO to R <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> ----- DB/EXPLAIN DB2=D42B----- CMD ==> SCROLL ==> PAGE P L A N F I L T E R I N G Cmds: GLOBAL (Menu) ----- FILTER ==> A (A AND R OR O OFF) OP VALUE OP VALUE PLAN PLAN NAME PQLF QUALIFIER PCR CREATOR PBBY BOUND BY PSRV SERVER PGRP GROUP MEMBER PBOT BIND TIME-YYYYMMDDHHMMSS OP VALUE OP VALUE PFRE FREED? (Y / N) PVLТ VALIDATE (B / R) PXPL EXPLAIN (Y / N) PISO = R ISOLATION (R / S / U) PDFP DEFERPREP (Y / N) PACQ ACQUIRE (U / A) POPR OPERATIVE (Y / N) PRLS RELEASE (D / C) PSQLR SQLRULES (D / S / -) PVLД VALID (Y / N / A) PDISC DISCONNECT (A / C / E / -) PDEGR DEGREE (1/ANY/-) PVRF VERIFIED? (Y / N / E) PEXPR EXPREDICATE (C/B/N) PDYNR DYNAMICRULES (R / -) PKEEP KEEP DYNAMIC (Y / N) PROPT REOPTVARS (Y / N) OP VALUE OP VALUE PSYEN (SYSENTRIES) PPLEN (PLENTRIES) PPSZ (PLSIZE) PCHSZ (CACHE SIZE) P#DP (ALL) PAVS (AVGSIZE) P#KA (PACKAGES) P#DB (DBRMS) ----- S Q L C O U N T S A L L S T M T S ----- (Filters applicable to Format 1 display) OP VALUE OP VALUE OP VALUE P#AL (ALTER) P#EC (EXECUTE) P#SE (SELECT) P#CA (CALL) P#EX (EXPLAIN) P#ST (SET) P#CL (CLOSE) P#FT (FETCH) P#SC (SET CON) P#CM (COMMIT) P#GR (GRANT) P#SD (SET DEG) P#CN (CONNECT) P#IN (INSERT) P#SH (SET HOS) P#CR (CREATE) P#LS (LOCK SHR) P#SP (SET PKG) P#DC (DCL CRS) P#LX (LOCK XCL) P#SR (SET RUL) P#DM (DCL STM) P#OP (OPEN) P#SS (SET SQL) P#DT (DCL TBL) P#PR (PREPARE) P#UP (UPDATE) P#DL (DELETE) P#RE (RELEASE) P#WH (WHENEVE) P#DS (DESCRIB) P#RV (REVOKE) P#TS (SEL+DCL) P#DR (DROP) P#RL (ROLLBACK) P#TL (TOTAL) ----- S Q L C O U N T S P A C K A G E S T M T S ----- (Filters applicable to Format 2 display) OP VALUE OP VALUE OP VALUE PK#AL (ALTER) PK#EC (EXECUTE) PK#SE (SELECT) PK#CA (CALL) PK#EX (EXPLAIN) PK#ST (SET) PK#CL (CLOSE) PK#FT (FETCH) PK#SC (SET CON) . . . </pre> </div>
6	Press Enter to save the entered values.

Process to change all plans with isolation level RR to CS (continued)

Step	Action
7	<p>Press End to see the filtered Plans panel. On that panel, type BINDREP on the command line. This creates the appropriate BIND commands with the replace option. The filtered plans will be rebound with the new options you specify on the Plan Bind Overrides panel.</p> <div data-bbox="573 541 1247 823" style="border: 1px solid black; padding: 5px;"> <pre> ----- DB/EXPLAIN DB2=D31A -----(F) CMD ==> BINDREP SCROLL ==> PAGE P L A N S Cnds: DO (Menu) GLOBAL (Menu) Selects: ? (Menu) ----- SEL PLAN COUNT STMTS SELECTS DCL CSR DELETES UPDATES INSERTS XCL SHR ----- - JNLPIPLN 1 2 0 0 0 2 0 0 0 - SUKPLAN 1 17 0 0 0 0 0 2 0 * TOTALS * ----- 36 0 1 0 2 2 0 ***** BOTTOM OF LIST ***** </pre> </div>
8	<p>If you have previously specified Y in the Display Override Panel for Command field of the Plan Bind Overrides Panel, !DB/EXPLAIN automatically displays it for you. Go to Step 10.</p>
9	<p>If you did not previously request display of Plan Bind Overrides panel, you must access this panel from the Housekeeping Menu. To do so:</p> <ul style="list-style-type: none"> ● On the command line of the filtered Plans panel, type HOUSE and press Enter. ● !DB/EXPLAIN displays the Housekeeping Menu. ● Select Plan Bind Overrides and press Enter.

Process to change all plans with isolation level RR to CS (continued)

Step	Action
10	<p>On the Plan Bind Overrides panel, type S in the Isolation field to change the isolation value to cursor stability from repeatable read. Type Y in the Use Overrides for Command? field.</p> <pre> ----- DB/EXPLAIN DB2=D42B ----- Cmd ==> P L A N B I N D O V E R R I D E S PERMANENT? ==> N (Y Yes N No) Use Overrides for Command? Y Select? N Convert? O Compare? N Display Override Panel for Command? Y Select? N Convert? N Compare? N Use Library Search Order for Command? Y Select? Y Convert? Y Compare? Y Owner : Action : (R Replace A Add) Qualifier : Retain : (Y Yes N No) Cache Size : Flag : (I / W / E / C) Current Server: Isolation: (R RR R RS S CS U UR) Current Data : (Yes N No) Acquire: (U Use A Allocate) SQLRULES : (D DB2 S STD) Release: (C Commit D Deallocate) Keepdynamic : (Y Yes N No) Validate: (R Run B Bind) REOPT(VARS) : (Y Yes N No) Explain : (Y Yes N No) Dynamicrules : (B / blank) Deferprep: (Y Yes N No) Gen PKLIST : (N No Y Yes) Degree : (1 / ANY) C Convert Only) Disconnect: (E Explicit A Automatic C Conditional) Collection Specification: Press ENTER to process. Press END to cancel. </pre>
11	<p>Press Enter.</p> <p>Result: If you so specified on the Output Options Defaults panel, !DB/EXPLAIN displays the Output Options panel. You can modify it by typing over any information you want to change.</p> <pre> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> O U T P U T O P T I O N S You have requested a function which will output data to a PDS member. Specify the PDS, member name, and whether to modify onto member or not. Library: TDDb.DE2.V230.USER Member: BIND Modify onto member: ENTER to accept values END to cancel request </pre>

Process to change all plans with isolation level RR to CS (continued)

Step	Action
12	Press Enter. Result: !DB/EXPLAIN displays the Process Member panel.
13	Type 1 in the Select Action field. <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> PROCESS MEMBER ----- DB/EXPLAIN DB2=D31A ----- BIND cmds saved Cmd ==> 99/10/06 07:57 Specify next action to be performed. Select Action ==> 1 1. UPDATE NOW Invoke DB2 to execute the member. 2. EDIT Edit the member. 3. DELETE Delete the member (and exit). 4. PRINT Print the member. 5. DO LATER Changes are saved, exit now. </pre> </div>
14	Press Enter. !DB/EXPLAIN processes the BIND Replace commands to change the isolation specification for the identified plans.

Change All Plans with an Isolation Level of RR (Repeatable Read) to CS (Cursor Stability)

Customizing !DB/EXPLAIN

Chapter 16. Controlling the Format of the EXPLAIN Display

Introduction

Generating EXPLAINS is a major function of !DB/EXPLAIN. Because each installation varies in its requirements for information, !DB/EXPLAIN provides you with predefined formats for the EXPLAIN display and also permits you to customize the format of the information you receive. For example, you can define what categories of information are shown to you in an EXPLAIN and specify the order in which they are presented. This chapter tells you how to control both the information that is gathered to support an EXPLAIN and the format of the EXPLAIN.

Chapter contents

Information Available in an EXPLAIN	304
Controlling What Information is Generated	306
Customizing Your EXPLAIN Output	311
Initial EXPLAIN Format	312
Changing the Initial Format During a Session or Batch Job	313
Changing the Format 0 Display	316

Information Available in an EXPLAIN

Overview

This unit tells you what categories of information are available in an EXPLAIN.

EXPLAIN predefined formats

The table below identifies the categories of information you can include in an EXPLAIN and maps the information category to the predefined format on which it appears.

Format Number	Provides information about...
1	The totality of the information categories contained on formats 2 through 9. Consult the remainder of this table.
2	PLAN_TABLE information. For each EXPLAINable statement, !DB/EXPLAIN displays information in summary form about the access path DB2 takes.
3	Object information. This format provides detailed object information for each EXPLAINable statement.
4	Bind information. This format provides detailed bind information for the plans or packages being EXPLAINed.
5	Statement information. This format provides the statement text for the EXPLAINable statements as well as the text related to any SQL errors.
6	Access path information. For each EXPLAINable statement, !DB/EXPLAIN displays information about the access path DB2 takes in narrative form.
7	Recommendation information. This format provides detailed object information for each EXPLAINable statement.
8	PLAN_TABLE information. For each EXPLAINable statement, !DB/EXPLAIN displays information about the access path DB2 takes in a 2-line format.
9	Statement cost information.

Format 0

In addition to the predefined formats listed above, you can generate your EXPLAIN in Format 0, a format whose contents and order of presentation you define completely. Information about defining Format 0 is found in the section “Defining Format 0” on page 311.

More information about an EXPLAIN

Refer to “EXPLAIN” in the *!DB/EXPLAIN Reference* for illustrations of an EXPLAIN in various formats.

Controlling What Information is Generated

Overview

Before you can generate an EXPLAIN display, you need to generate the information from which the display is populated. This unit describes how you can use the various fields on the EXPLAIN Display Tuning Parameters panel to control or limit the information that comprises an EXPLAIN.

EXPLAIN Display Tuning Parameters panel

You specify what information is to be generated using Housekeeping's EXPLAIN Display Tuning Parameters panel. An annotated copy of this panel follows.

```

----- DB/EXPLAIN DB2=D31A -----
Cmd ==>

      E X P L A I N   D I S P L A Y   T U N I N G   P A R A M E T E R S

PERMANENT? ==>   ( Y Yes  N No )

                3          7          6
                Generate?  Format0    Order
                Display?
EBIND information      Y          N          1
ERECM (Bind Specific) 4        Y          3 1
ECOST information      Y          N          2
EOBJ information       Y          N          5
EKEYS information 5        Y          N
EPATH information      Y          N          3
EPTBL information      Y          N          2
EPTB2 information      Y          N          2
ERECM information      Y          N          4
ESTMT information      Y          N          1

ENTER to process  END to cancel
    
```

EXPLAIN Display Tuning Parameters panel (continued)

The list below describes various fields on the display. Each field is discussed further later in this chapter.

- 1** Parameters that define the BIND section of the EXPLAIN display
- 2** Parameters that define the Statements section of the EXPLAIN display
- 3** Specification that determines whether or not information is generated
- 4** ERECM parameter in the BIND section does not have a generation option
- 5** Generation of EKEYS requires that you specify Y for the generation of EOBJ.
- 6** Specification of the order in which information appears on the EXPLAIN display
- 7** Specification for the presence of information on a Format 0 display

Accessing the EXPLAIN Display Tuning Parameters panel

You can access the EXPLAIN Display Tuning Parameters panel from the Housekeeping panel as described in “EXPLAIN Display Tuning Parameters” in the *!DB/EXPLAIN Reference*, or by entering **HOUSE XTU** on the command line of any *!DB/EXPLAIN* panel. **XTU** is the mnemonic for this Tuning Parameters panel.

Defining sections of the EXPLAIN display

The EXPLAIN display contains two sections: the BIND section and the statements section. On the EXPLAIN Display Tuning Parameters panel, the first three parameters (**1**) define the BIND section; the final six parameters (**2**) define the Statements section.

Parameters specify categories of information

The table below identifies the parameters that represent categories of information you want to generate and maps them to categories of EXPLAIN information.

Parameter	Generates
EBIND	Plan bind or package bind information
ECOST	Cost information
EOBJ	Detailed object information
EKEYS	Index key information
EPATH	Access path summary information in textual form
ERECM	Recommendations. This parameter appears twice on the panel, the first time related to BIND statements, the second time related to statements in general. Additional information on limiting the specification of ERECM appears below.
ESTMT	Statement text is generated. If !DB/EXPLAIN encounters an SQL error when preparing the statement to be EXPLAINed, then SQL error text is also generated.
EPTBL and EPTB2	PLAN_TABLE information in a one-line or two-line format, respectively.

Generating information for a category

To cause !DB/EXPLAIN to generate a specific category of information, type **Y** in the corresponding Generate? field (**3**) of the EXPLAIN Display Tuning Parameters panel. In order to generate EKEYS information (**5**), you must specify **Y** for EOBJ.

ERECM—an exception

The ERECM BIND specific parameter (**4**) does not have a generate option; it uses the generation option you specify for ERECM in the statements section of the panel.

Limiting the information you generate

To limit the number of display lines !DB/EXPLAIN generates, type **N** in the corresponding Generate? field (**3**) of the EXPLAIN Display Tuning Parameters panel.

Limiting the information generated allows you to control the amount of storage required to generate EXPLAINS.

Remember that only the display lines you generate can be displayed.

Changing what you generate

If you want to view display lines you have not previously generated, follow the steps in the table below.

Step	Action
1	From the Housekeeping panel, select EXPLAIN Display Tuning Parameters.
2	Press Enter. Result: The EXPLAIN Display Tuning Parameters panel displays.
3	On this panel, review your current selections in the Generate? column. Respecify those selections required to make corresponding changes to your next EXPLAIN and press Enter.
4	Return to the Primary Menu. Execute another EXPLAIN or, to save storage, a latest EXPLAIN. Result: !DB/EXPLAIN displays the EXPLAIN that reflects your altered generate options.

Structure of your EXPLAIN display output

The EXPLAIN that is generated is structured as follows:

 BIND section for package (or plan) #1

 Statement section for statement #1

 Statement section for statement #2

 ⋮

 Statement section for statement #n

 BIND section for package (or plan) #2

 Statement section for statement #1

 Statement section for statement #2

 ⋮

 Statement section for statement #n

 BIND section for package (or plan) #3

 ⋮

Customizing Your EXPLAIN Output

Overview

You can customize the order in which information is presented to you on your EXPLAIN display. You can also customize an entire display format, called Format 0. This unit will explain how.

Defining the order of displayed information

On the EXPLAIN Display Tuning Parameters panel, use the column called Order to arrange the presentation of information. The order you specify is the order in which the information appears in the section of the EXPLAIN display to which it is applicable.

- Order the BIND section parameters by numbering them 1 through 3.
- Separately order the Statements section parameters by numbering them 1 through 5.

Specify an order number for EPTABL or EPTAB2, but not both.

Note: You specify order as if all of the parameters in a given section will be displayed. Thus, even though you specified Generate? N for the EOBJ parameter, you must still give the EOBJ parameter an ordering number.

Defining Format 0

To create a Format 0 display, use the Format0 Display? column. (The Format0 Display column on the EXPLAIN Display Tuning Parameters display replaces the “toggle switch” on the Panel Formats panel. The toggle switch was used to perform the format 0 definition prior to Version 235.) This column permits you to tailor a default Format 0 display. Type Y for those parameters whose information you want to appear on the display. Type N for those parameters whose information you want to omit from the display.

Remember, if you specify Format 0 as Y for a parameter whose Generate? specification is N, you have requested that this information be included in your EXPLAIN. However, no information can be displayed because it was not generated.

There are some special considerations when working with a Format 0 display. See the section “Changing the Format 0 Display” on page 316 for additional information.

Initial EXPLAIN Format

Overview

This section describes how !DB/EXPLAIN determines what format to display when you request an EXPLAIN in an online session or batch execution.

Specifying the initial display

To specify the initial format for the EXPLAIN display, follow the steps in the chart below.

Step	Action
1	From !DB/EXPLAIN's Housekeeping panel, select Panel Formats. Result: The Panel Formats panel is displayed
2	On the line for the EXPLAIN display, in the Format # column, enter the number of the format you intend to be the initial format.
3	Press Enter. Result: When you request an EXPLAIN: <ul style="list-style-type: none"> • In an online session, the initial display you receive is in the format you specified in Step 2. • In a batch session, the initial display you receive is in the format you specified in Step 2 unless you have overridden the specification using the KTECNTL input job stream. See "Overriding the initial display in batch" below.

Discontinuance of the toggle switch

In versions of !DB/EXPLAIN prior to Version 235, a "toggle switch" was included on the Panel Formats panel. This toggle switch permitted you to define to !DB/EXPLAIN the types of information you wanted on your Format 0 display. The toggle switch has been replaced by the Format 0 Display? column on the EXPLAIN Display Tuning Parameters panel. See "Defining Format 0" above for information about the use of this column.

Changing the Initial Format During a Session or Batch Job

Overview

This unit describes the procedures to change the format of an EXPLAIN while in an online session or batch execution.

Overriding the initial display in batch

The format of the EXPLAIN initially displayed in a batch run is the format you have specified on the Panel Formats panel unless you include an override in your KTECNTL batch input stream. When running a batch job, use the KTECNTL parameter **EXP_EX_FORMAT=** to override the initial specification you made on the Panel Formats panel (see Example 2 below).

If you want to change the format several times within the batch execution, include the **FORMATn** command in your batch input stream (see Example 3 below).

Example 1

The value in the Panel Formats Format # column is 3. An override command is *not* included in the input job stream.

```
//SYSIN DD *  
:  
EXPLA  
PRNT  
END  
/*
```

Result: The format of the EXPLAIN sent to the report output device is Format 3.

Example 2

The value in the Panel Formats Format # column is 3. An override command is included in the input job stream.

```
//KTECNTL DD *
  EXP_EX_FORMAT=2
/*
//SYSIN DD
  EXPLA
  PRNT
  END
/*
```

Result: The format of the EXPLAIN sent to the report output device is Format 2.

Example 3

The value in the Panel Formats Format # column is 3. An override command is included in the input job stream. An additional format command is included later in the job stream to change the format again.

```
//KTECNTL DD *
  EXP_EX_FORMAT=2 (Override Panel Formats panel specification)
/*
//SYSIN DD *
  EXPLA
  PRNT (Output EXPLAIN in format 2)
  FORMAT4 (Change to Format 4)
  PRNT (Output EXPLAIN in format 4)
  END
/*
```

Result: The format of the first EXPLAIN sent to the report output device is Format 3; the format of the second EXPLAIN is Format 4.

Changing the display during an online session

In an online session, after the initial EXPLAIN panel is displayed (according to the value you set on the Panel Formats panel), you can use the `FORMATn` command to view displays in alternate formats.

Step	Action
1	On the command line of the EXPLAIN display, enter the command <code>FORMATn</code> , where <i>n</i> is a value from 0 through 9.
2	Press Enter. Result: !DB/EXPLAIN displays the EXPLAIN in the format you specified in the command.

Changing the Format 0 Display

Overview

This section discusses the commands specific to the Format 0 display and how these commands can be entered in batch and online sessions.

Format 0 commands

The table below shows the commands you can enter either from the command line of any EXPLAIN display or within a SYSIN DD command grouping to affect the Format 0 display.

Command	Displays...
EBIND	Plan bind or package bind information
ECOST	Cost information
EKEYS	Index keys information
EOBJ	Detailed object information
EPATH	Access path summary information in textual form
ERECM	Recommendations
ESTMT	Statement text and SQL error text (for SQL errors encountered)
EPTBL and EPTB2	PLAN_TABLE information in a one-line or two-line format, respectively.
EALL	Display all sections
RESET	No sections. !DB/EXPLAIN clears all previous selections and prepares your session so that you can add back sections of the display you want to see by using the commands above.

Notes on the use of Format 0 commands

- You may request that a section be displayed; however, if information for that section has not been generated, no information is displayed.
- In an online session, you can enter the format 0 commands from any EXPLAIN display format command line. However, the commands are not effective unless you are displaying Format 0.
- When these commands are input at the command line, they affect what information is displayed. The order in which the information is displayed is the order you initially specified on the EXPLAIN Display Tuning Parameters panel.
 - In a batch job, override the default order by including profile variables **EXP_EXORD_commandname** in your KTECNTL job. The example below causes the object information to appear before the cost information:

```
// KTECNTL DD *
  EXP_EXORD_EBIND=1
  EXP_EXORD_ECOST=3
  EXP_EXORD_EOBJ=2
/*
```

- In an online session, you can override the order for the session by modifying the Order? field of the EXPLAIN Display Tuning Parameters panel.

Chapter 17. Controlling the Destination of !DB/EXPLAIN Output

Introduction

This chapter contains information about controlling the destination of output created by !DB/EXPLAIN.

Chapter contents

Using Output Options	320
Using the Count Variable on Commands and Selects	322
Using the MOD Option on Commands and Selects	325
Viewing Profile Datasets	326
Setting DB/Tools Profile Datasets	328

Using Output Options

Overview

This unit describes the use of the Output Options Defaults and Output Options panels to control the destination of output created by !DB/EXPLAIN.

Specifying default destinations for your output

Follow these steps to specify a default destination for output created by !DB/EXPLAIN.

Step	Action
1	On the Housekeeping Menu panel, select Output Options.
2	Press Enter. Result: !DB/EXPLAIN displays the Output Options Defaults panel
3	In the Member Name column, type the name of a member in the user PDS where the output of this function is to be stored.
4	To make the entries permanent (if you are appropriately authorized), type Y in the Permanent field.
5	Press Enter. Result: !DB/EXPLAIN makes the entries on this display permanent.

Specifying a destination for output at execution time

If you think you will want to make a decision about the destination of a given output at execution time, follow the steps in this table.

Step	Action
1	On the Output Options Defaults panel, type Y in the Display Panel field.
2	Press Enter. Result: !DB/EXPLAIN displays the Output Options panel to permit you to change the member to which the output is written. Your specification on the Output Options panel overrides the default value you specified on the Output Options Defaults panel.

Format of the commands that create output

The commands that write their output to a PDS member can be used alone, for example, **OUT** or **BIND**. Alternatively, a command can take the form *commandname member*, for example, **OUT MYMEMBER**.

When you use the command alone, **!DB/EXPLAIN** writes the output of the command to the member whose name you specified on Housekeeping's Output Options Defaults panel in the Member Name field.

When used with a member name, *member* identifies the member in the user PDS to which **!DB/EXPLAIN** writes the output. This specification overrides the specification you made on the Output Options Defaults panel.

Using the Count Variable on Commands and Selects

Overview

This unit describes how you can use the count variable (&CNT) when writing output to a partitioned dataset using a !DB/EXPLAIN command or select.

Background about the &CNT variable

When you issue a command or select to write output to a PDS member, that output overwrites the current contents of the member (unless you specified MOD in the command or as a default. See the information about MOD later in this chapter.) Using the &CNT variable ensures that a string of commands and selects that output information write to a new member each time they are issued.

How is &CNT specified?

To use &CNT, you append the variable &CNT to the member name you specify in the Member Name column of Housekeeping's Output Options Defaults panel. The name of the member can be no longer than 4 characters and the total length of the member name plus the counter cannot exceed 8 characters.

What is the &CNT variable?

&CNT is an 8-position counter. Each time you start a !DB/EXPLAIN session, &CNT is set to 0. Each time you issue a command that generates output to a user PDS member during a !DB/EXPLAIN session, !DB/EXPLAIN increments &CNT by 1 and appends the value to the member name.

Example of member names used with &CNT

In this example, member names identified by **1** illustrate the use of the &CNT variable.

```

----- DB/EXPLAIN DB2=D31A -----
Cmd ==>

      O U T P U T   O P T I O N S   D E F A U L T S
PERMANENT? ==> N ( Y Yes N No )

      DISPLAY      MEMBER      MOD ONTO
      PANEL?      NAME         MEMBERS?
-----
BIND          Y          BIND&CNT  1          N
DROP          Y          FREE      N
EXPLAIN      Y          EXPLAIN  N
JCL          Y          JCL      N
OUT          Y          SQL&CNT  1          N
REBIND       Y          REBIND  N
STATS       Y          STATS   N

      BIND COMPARE      COMPARE
      HOST VARIABLES    HOSTS

ENTER to process  END to cancel

```

Example of adding a numeric value to a member name to create a new member

Here is an example of how &CNT is used when you issue a sequence of commands and selects during a &EXP. session. The example reflects the selections shown on the panel above.

Issue This Command or Select	On This Panel	To Generate This PDS Member
OUT command	Statements panel	SQL00000
OUT command	Statements panel	SQL00001
O select	Statements panel	SQL00002
B select	Plans panel	BIND0003

Command and selects with which you can use &CNT

These are the commands and selects to which you can attach the &CNT variable.

Commands	Selects
BIND, BINDADD, BINDCNV, BINDEXP, and BINDREP	B
DROP	F
EXODS	
FREE	F
JCL	
OUT (SQL and STATISTICS)	O
REBIND and REBINDINDEX	R

Format of the commands that use &CNT

You can use the &CNT variable in association with commands entered from the command line of an object list panel. Commands that use the &CNT variable take the form:

commandname member&CNT

When used in this form, *member* represents the name of a member in the user PDS. !DB/EXPLAIN appends the counter value to the member name to ensure that previous members with the same member name are not overwritten.

Using the MOD Option on Commands and Selects

Overview

This unit describes how you can use the MOD option when writing output to a partitioned dataset using a !DB/EXPLAIN command or select.

Effect of using MOD

When a member is identified as using MOD, any output written to that existing member is appended at the end of any data already in the member.

Specifying whether or not MOD is used

To specify that MOD is to be used for a given output type, follow the steps in this table.

Step	Action
1	On the Output Options Defaults panel, type Y in the MOD Onto Members? field.
2	Press Enter. Result: !DB/EXPLAIN now appends output to the specified datasets rather than overwriting the existing input.

Command and selects with which you can use MOD

These are the commands and selects to which you can attach the MOD option.

Commands	Selects
BIND, BINDADD, BINDCNV, BINDEXP, and BINDREP	B
DROP	F
EXODS	
FREE	F
OUT (SQL and STATISTICS)	O
REBIND and REBINDX	R

Format of the commands that use MOD

Commands that use the MOD option take the form:

- *commandname,MOD*

When used in this form, the output of the command is appended to the end of the data in the default member. The default member is the name you specify on Housekeeping's Output Options Defaults panel in the field Member Name. Using MOD explicitly on the command overrides your specification in the Use MOD? field on the Output Options Defaults panel.

- *commandname member,MOD*

When used in this form, *member* represents the name of a member in the User PDS to which !DB/EXPLAIN appends the command output. Explicitly specifying MOD on the command overrides your specification in the Use MOD? field on the Output Options Defaults panel.

Viewing Profile Datasets

Overview

This unit describes the SQL Formats for KTEXPL panel. By setting up profiles, you can place restrictions on specified users and groups.

Accessing the DB/Tools Allocated Profile Data Set panel

Use the following procedures to display information about profile datasets.

Step	Action
1	On the Administration Menu, type 7 in the Option field.
2	Press Enter. Result: !DB/EXPLAIN displays the DB/Tools Allocated Profile Data sets panel.

Elements of the DB/Tools Allocated Profile Data sets panel

This is the DB/Tools Allocated Profile Data sets panel.

```
----- DB/Tools -----  
Cmd ==>  
  
          DB/Tools Allocated Profile Data sets  
  
User   : 1  
Group  : 2  
  
System : TDKT.DBT00LS.E500.PROFILE 3
```

- 1** The name of the user for whom you are setting a profile. Typing a select in the SEL column next to a member name causes !DB/EXPLAIN to display all of the individual members of the logical group the member name represents.
- 2** The name of the group that the user belongs to whose profile you are setting up.
- 3** The system where the profile resides, and from which you access and store all profile information.

Setting DB/Tools Profile Datasets

Overview

This unit describes the how to set user and group profiles on the DB/Tools Profile Information panel. Use the fields on this panel to store user defaults and values and to provide for authorization to change these defaults.

Background about the DB/Tools Information panel

The DB/Tools Information panel can be used to store specific information about a user or group. From here, you provide authorization so that a variety of functions either can or cannot be carried out from here. Default values permits users to make or to not make certain changes depending on how the profile was set up.

Using the DB/Tools Profile Information panel

Use the following procedures to set user and group profile datasets.

Step	Action
1	On the Administration Functions panel, type 8 in the Option field.
2	Press Enter. Result: !DB/EXPLAIN displays the DB/Tools Profile Information panel.

Elements of the DB/Tools Profile Information panel

This is the DB/Tools Profile Information panel.

```

----- DB/Tools -----
Cmd ==>

                DB/Tools Profile Information

Save Values? 1          ==> Y ( Y Permanent  N Temporary
Display this panel on product entry? ==> N ( Y Yes  N No ) 2
3
Use? Type 4      Profile Data Set Name 5      Status 6
-----
N User : 7 _____
N Group: 8 _____
N
N
N
N
N
N
N
N
Y      System: TDKT.DBTOOLS.E500.PROFILE      OK

Press ENTER to continue. Press END to exit.

```

- 1** Specify if you want values to be stored after exit. Values are **Y** for permanent storage or **N** for storage to last as long as you are in this file. Default is **Y**.
- 2** Specify if you want the profile panel to display whenever the product is first opened. Values are **N** or **Y**. Default is **N**.
- 3** Use default is **N**.
- 4** Enter the type for the user and group.
- 5** Enter the name of the profile dataset.
- 6** The status of the profile.
- 7** User information displays here.
- 8** Group information displays here.

The Extract Program

Chapter 18.

The !DB/EXPLAIN Extract Program and the Extract ID

Introduction

An efficient and flexible !DB/EXPLAIN batch extract allows you to tailor the extract to your site and to do incremental extracts. Using the extract helps conserve system resources while keeping the extract as current as the needs of your installation require. This chapter tells you how to create an extract, create extract IDs, and limit the data you extract.

Chapter contents

Definition and Features of an Extract	334
Background about the Extract ID	336
Viewing Available Extract IDs	338
Selecting an Extract ID to Work With	341
Defining the Extract ID	342
How Many Extract IDs Should You Define?	344
Customizing Extract IDs in Your System Profile dataset ..	346
Extract Masking	348

Definition and Features of an Extract

Overview

This unit defines !DB/EXPLAIN's batch extract and describes its major features.

Definition of an extract

The batch jobs that extract data about plans and packages from the DB2 Catalog or from a DB2 PLAN_TABLE and place the data in VSAM KSDS(s) collectively are called *the extract*.

Extract data is that data stored in the extract VSAM dataset(s) as a result of running an extract.

Features of the !DB/EXPLAIN extract

The list below describes the major features of !DB/EXPLAIN's extract.

- !DB/EXPLAIN's extract derives its data from DB2's catalog tables. Once an extract is performed, you can access the extracted data an infinite number of times without impacting the active DB2 system. The advantages of using extract technology rather than direct catalog access are:
 - You can access extract data even when DB2 is not active.
 - You avoid contention on the DB2 catalog.
- !DB/EXPLAIN's extract technology provides an advantage over the use of DB2 shadow catalog tables in that once the extract is performed, the data is available without regard to the availability of your DB2 subsystem.
- You can extract historical EXPLAIN data from a user-specified PLAN_TABLE.
- You can retain histories of plan, package, and DBRM catalog information, and their EXPLAINS. You can then compare a historical plan, package, or DBRM to its current version to assist in diagnosing performance changes after a BIND/REBIND.
- You can limit the extracted data.
 - Execution parameters allow you to extract only that catalog data relating to plans or packages bound since the last extract.
 - Masking allows you to limit the plans or packages selected.
- !DB/EXPLAIN stores extracted DBRMs, plans, and statements independently by type. This permits application-based extracts to be performed; that is, extracts can be based on business function, area of responsibility, development vs. production, or another subset of your choosing.
- You can extract and combine data from multiple PLAN_TABLES to meet your information requirements.
- !DB/EXPLAIN sorts extract data external to DB2. This process is less expensive in terms of system resources than performing an SQL ORDER BY against a DB2 table.
- !DB/EXPLAIN stores the extracted data into a VSAM Key Sequenced dataset(s) (KSDS).

Background about the Extract ID

Overview

This unit provides background information about the !DB/EXPLAIN extract ID.

Definition of an extract ID

The extract ID is a unique 1- to 8-character identifier that identifies the extract. It is the basic configuration unit for !DB/EXPLAIN and identifies an extract or set of extracts performed on a DB2 subsystem. Several extracts can be stored in the same dataset. You can define the extract ID to identify all data extracted from a DB2 subsystem, or you can associate the extract ID with a particular range or set of data extracted from that subsystem. These concepts are explored further in “How Many Extract IDs Should You Define?” on page 344.

Relationship of !DB/EXPLAIN's extract ID to the DB2 subsystem ID

Candle suggests that the first characters of the extract ID be the DB2 subsystem ID to make for easier tracking of extracts. The DB2 subsystem ID is a 1- to 4-character ID identifying the DB2 subsystem.

A given extract ID is associated with a specific DB2 subsystem. It determines the DB2 subsystem from which data is obtained and where the data is placed—a specific extract ID in a specific VSAM cluster. It also controls subsequent access from online !DB/EXPLAIN.

Relationship of !DB/EXPLAIN's extract ID to the MVS ID

The MVS ID is a 3- to 8-character ID that specifies the MVS system. One or more DB2 subsystems can be associated with a given MVS subsystem. Similarly, by being associated with a specific DB2 subsystem ID, an extract ID is associated with that DB2 subsystem's MVS system ID for documentation purposes.

The same DB2 subsystem ID can exist on multiple MVS systems at the same site. Thus, the association of extract ID to MVS ID maintains the uniqueness of the extract data.

If DASD is shared between MVS systems, multiple extract IDs can share the same VSAM clusters and run from the same system profile data set. The advantage to this is that less customization and administration are required to manage your extracts.

Extract ID for online sessions

CLISTs, EXECs, and edit macros (for example, KTEXPL, KTEXPLA, KTEQMF, KTEXPLD, and KTEXPLB) that do not use extract data, but are invoked during an online !DB/EXPLAIN session, determine which extract ID to use as follows:

- If you have not specified the extract ID to be used, !DB/EXPLAIN retrieves the extract ID from the !DB/Tools Product Profile.
- If an extract ID is not found in the !DB/Tools Product Profile, !DB/EXPLAIN displays an error message.

Viewing Available Extract IDs

Overview

This unit provides information about viewing available extract IDs.

Background about viewing the available extract IDs

If you want to use an extract ID other than the one stored in your !DB/Tools Product Profile, you may want to view the extract IDs available to you. To view the extract IDs for which your !DB/Tools are configured, your actions will vary depending on the panel from which you access !DB/EXPLAIN.

When entering !DB/EXPLAIN from the !DB/EXPLAIN Primary Menu

To view the extract IDs available to you when entering !DB/EXPLAIN from the Primary Menu, follow these steps.

Step	Action
1	On the !DB/EXPLAIN Primary Menu, enter ? in the Extract ID field.
2	Press Enter. Result: !DB/EXPLAIN displays the !DB/Tools Global Information—DB2 Subsystem Name Table panel. On this panel, extract IDs you can access are mapped to DB2 subsystem IDs and MVS IDs.
3	To select an extract ID to work with, follow the steps provided in “Selecting an Extract ID to Work With” on page 341.

Note: An alternative to entering ? on the Primary Menu is to use the DB2 IDs field on the Administration Menu. However, using the question mark in the Extract ID field is the most direct method.

When entering !DB/EXPLAIN from the !DB/Tools Product Selection Menu

To view the extract IDs available to you when you access !DB/EXPLAIN from the !DB/Tools Product Selection Menu, you can either leave the Extract ID field blank or enter an extract ID, even an invalid one. The tables that follow describe the results of these choices.

Leaving the Extract ID field blank

Here is the process to view an extract ID when accessing !DB/EXPLAIN from the !DB/Tools Menu, leaving the extract ID blank.

Step	Action
1	On the !DB/Tools Product Selection Menu, select !DB/EXPLAIN and leave the Extract ID field blank.
2	Press Enter. Result: You receive the !DB/EXPLAIN Primary Menu with NONE in the Extract ID field.
3	On the !DB/EXPLAIN Primary Menu, type ? in the Extract ID field.
4	Press Enter. Result: !DB/EXPLAIN displays the !DB/Tools Global Information—DB2 Subsystem Name Table panel. On this panel, extract IDs you can access are mapped to DB2 subsystem IDs and MVS IDs.
5	To select an extract ID to work with, follow the steps provided in “Selecting an Extract ID to Work With” on page 341.

Entering any extract ID in the Extract ID field

This table describes the process for viewing available extract IDs when accessing !DB/EXPLAIN from the !DB/Tools Primary Menu and you enter an extract ID, even an invalid one, in the Extract ID field.

Step	Action
1	On the !DB/Tools Product Selection Menu, select !DB/EXPLAIN and enter any extract ID in the Extract ID field. The extract ID you enter can be either valid or invalid.
2	<p>Press Enter.</p> <p>Result: You receive the !DB/EXPLAIN Primary Menu with the value you entered in the Extract ID field of the !DB/Tools Product Menu displayed in the Extract ID field of the !DB/EXPLAIN Primary Menu.</p> <ul style="list-style-type: none"> ● If this is a valid extract ID and is the one you want to work on, press Enter again to continue with your !DB/EXPLAIN session. ● If this is a valid extract ID, but you want to view additional available extract IDs, continue to Step 3. ● If this is an invalid extract ID, !DB/EXPLAIN displays the message NOT CONFIGURED in the upper right corner of the Primary Menu panel. Continue to Step 3.
3	On the !DB/EXPLAIN Primary Menu, enter ? in the Extract ID field.
4	<p>Press Enter.</p> <p>Result: !DB/EXPLAIN displays the !DB/Tools Global Information—DB2 Subsystem Name Table panel. On this panel, extract IDs you can access are mapped to DB2 subsystem IDs and MVS IDs.</p>
6	To select an extract ID to work with, follow the steps provided in “Selecting an Extract ID to Work With” on page 341.

Selecting an Extract ID to Work With

Overview

This unit describes how you can select an extract ID to work with when accessing !DB/EXPLAIN.

Background about selecting the extract ID

This unit assumes that you have left the extract ID field blank when accessing !DB/EXPLAIN, or the value you entered in the extract ID field is invalid. Alternatively, you arrived at this point by entering ? in the Extract ID field on the Primary Menu or selecting **DB2 IDs** on the !DB/EXPLAIN Administration Menu.

Selecting the extract ID to use

Follow the steps in this table to select an extract ID to work with.

Step	Action
1	On the !DB/Tools Global Information—DB2 Subsystem Name Table panel, select a specific extract ID to work on by typing any character in the selection column.
2	Press Enter. Result: !DB/EXPLAIN returns you to the !DB/EXPLAIN Primary Menu, placing the extract ID you selected into the Extract ID field on that panel.
3	Press Enter again to continue with your !DB/EXPLAIN session.

Defining the Extract ID

Overview

This unit discusses how an initial extract ID is defined and how additional extract IDs can be added to !DB/EXPLAIN.

Initial extract job

When you install !DB/EXPLAIN for the first time, the installation process creates an initial extract job called *db2idJXTE* and places it in the CNTL PDS.

Defining the initial extract ID

When you execute JCL member *db2idJXTE*, you specify the extract ID to be assigned to the current extract as part of the SYSIN input job stream.

Restriction on adding extract IDs

You cannot set up additional extract IDs on a DB2 subsystem on which you have not installed !DB/EXPLAIN.

Procedure for adding an extract ID

After you have installed !DB/EXPLAIN on a given DB2 subsystem, follow this procedure to set up additional extract IDs on that subsystem.

Step	Action
1	Start !DB/EXPLAIN by typing the following on the command line of an ISPF panel: <pre>TSO %KTE <i>extract_id</i> SYSPROF(<i>hilev.PROFILE</i>) USERPROF(<i>userid.PROFILE</i>) AUXPROF(<i>auxid.PROFILE</i>)</pre>
2	Press Enter. Result !DB/EXPLAIN displays the Primary Menu together with a message telling you that the extract ID is not configured.
3	On the Primary Menu, type H0US on the command line.
4	Press Enter. Result: !DB/EXPLAIN displays the Housekeeping Menu.

Procedure for adding an extract ID (continued)

Step	Action
5	On the Housekeeping Menu, select Global Information.
6	Press Enter. Result !DB/EXPLAIN displays the !DB/Tools Global Information panel. Review this panel to verify the values it contains. Type any values you want to change over the displayed values.
7	Press Enter Result !DB/EXPLAIN displays the Housekeeping Menu.
8	On the Housekeeping Menu, select DB2 Subsystem Information.
9	Press Enter. Result !DB/EXPLAIN displays the !DB/Tools DB2-Specific Information panel. Verify the values on this panel. Type any values you want to change over the displayed values.
10	Press Enter. Result !DB/EXPLAIN redisplay the Housekeeping Menu.
11	On the Housekeeping Menu, select !DB/EXPLAIN Configuration.
12	Press Enter. Result !DB/EXPLAIN displays the !DB/EXPLAIN Configuration Information panel. Verify the values on this panel. Type any values you want to change over the displayed values.
13	Press Enter. Result !DB/EXPLAIN redisplay the Housekeeping Menu. The extract ID is now configured.
14	Submit the job to run the new extract.

Propagating !DB/EXPLAIN to additional subsystems

If you want to propagate your !DB/EXPLAIN to additional subsystems, use the !DB/Tools ADDSYS install option. The install process creates unique DB2 extract jobs for those subsystems by concatenating the DB2 subsystem ID with the letters *JXTE*. For additional information about the ADDSYS option, see the *!DB/Tools Installation and Customization Guide* section entitled “The Installation Process.”

How Many Extract IDs Should You Define?

Overview

This unit discusses the advantages and disadvantages associated with the number of extract IDs you define.

Background about defining one or multiple extract IDs

The number of extract IDs you define per DB2 subsystem is flexible. Based on the needs of your installation, you can define only one extract ID per DB2 subsystem or you can define many.

One extract ID

When you have specified only one extract ID per DB2 subsystem, that extract ID contains all relevant data from the DB2 catalog. The advantages of defining only one extract ID are:

- Your !DB/EXPLAIN system requires less customization.
- Your administration of extract IDs is potentially less complex.

The disadvantages of defining only one extract are:

- Your extract runs are longer.
- You can experience more access conflicts
- You can be less flexible in the tasks you can accomplish using the extract data.

Many extract IDs

You can use extract masking to set up multiple data groupings, each having its own unique extract ID. The advantage to defining many extract IDs is that you can set up and run extracts by one or more categories that fit your installation, such as: organization, product line, area of responsibility, or production versus development. The advantages of defining multiple extract IDs per DB2 subsystem are:

- Your extract runs are shorter
- You experience fewer access conflicts

The disadvantages of defining many extract IDs are:

- More customization of your !DB/EXPLAIN system is required.
- More complex operational activities may be required.
- Multiple extracts cannot be combined to permit you to view data as if from one extract.

Customizing Extract IDs in Your System Profile Dataset

Overview

This unit discusses how the !DB/EXPLAIN system profile dataset stores the values associated with a given extract ID. It provides high-level information on profile datasets.

What is a profile dataset?

You use profile datasets to tailor your !DB/EXPLAIN system and user defaults to those that make the most sense for your installation. Profile datasets contain configuration information and product defaults. Profile datasets are initially set up during !DB/EXPLAIN installation and can be modified either temporarily or permanently by means of !DB/EXPLAIN's housekeeping options.

If DASD is not shared between MVS systems, each MVS system has its own system profile dataset. MVS systems that share DASD can run from the same system profile data set. Each extract ID has its own set of members stored in the system profile dataset. The extract ID members contain the defaults for that extract ID.

Authorization

In this and the following units of this chapter, you are told that “!DB/EXPLAIN writes the new values to the profile dataset...” This is only true if you have authorization to write to the profile dataset. You *cannot* make changes permanent without appropriate authorization.

Procedure for customizing your extract IDs

The table below provides instructions for customizing (sometimes called configuring or reconfiguring) your extract IDs:

Step	Action
1	On !DB/EXPLAIN's Housekeeping Menu panel, select !DB/EXPLAIN Configuration.
2	Press Enter. Result: !DB/EXPLAIN displays the Configuration Information panel.
3	On the !DB/EXPLAIN Configuration Information panel, review the plan names, product dataset names (system, user, log, and extract partitioned datasets), and output characteristics associated with a given extract ID.
4	Type new values for any fields you want to change over the displayed values.
5	Press Enter. Result: !DB/EXPLAIN writes the new values you entered to the profile dataset and redisplay the Housekeeping Menu panel.
6	On !DB/EXPLAIN's Housekeeping Menu panel, select DB2 Subsystem Information.
7	Press Enter. Result: !DB/EXPLAIN displays the !DB/Tools DB2-Specific Information panel to display information associated with a given extract ID. This information includes: <ul style="list-style-type: none"> • The DB2 subsystem ID for the extract ID taken from the extract ID profile. The extract ID identifies the release of DB2 associated with it. • The name of the DB2 DSNLOAD dataset for the extract ID. The DSNLOAD for the extract must be compatible with the release of DB2 identified by the extract ID.
8	On the !DB/Tools DB2-Specific Information panel, review the displayed information.
9	Type new values for any fields you want to change over the displayed values.
10	Press Enter. Result: !DB/EXPLAIN writes the new values you entered to the profile dataset and redisplay the Housekeeping Menu panel.

Extract Masking

Overview

This unit discusses how you can limit extracted data by using masks.

Limiting extracted data

You can limit the amount of data that is extracted from the DB2 Catalog or PLAN_TABLE by using !DB/EXPLAIN's batch extract masking and by specifying extract function options on an extract run. Extract function options are discussed in the unit “Specifying function options” on page 360.

Batch extract masking

When using !DB/EXPLAIN's batch extract masking, you specify timestamp, plan, and/or package masks to selectively extract DB2 Catalog data. You specify plan and/or package masks to selectively extract PLAN_TABLE data. These masking criteria define the range of plans and packages you want. The masks are specified using the plan specification, package specification, and, if applicable, timestamp specification parameters on SYSIN input statements in your extract batch job. See “Input Syntax” on page 370 for the syntax of your SYSIN input.

If you do not specify masking, the extract job defaults to extracting all plans and/or all packages.

Remember that in the batch extract, masking *must be specified separately* for plans and for packages.

Online masking

During an online execution, you can determine whether or not masking is in effect by examining the Masking in Effect? field on !DB/EXPLAIN's Primary Menu. The field can contain either Y (yes) or N (no).

When masking is in effect, !DB/EXPLAIN displays the letter M in parentheses in the upper right hand corner of object list panels. If both masking and filtering are in effect, !DB/EXPLAIN displays the letters FM. Online masks provide an additional layer of masking on top of the batch extracted data.

Syntax for PLAN, PACKAGE, and TIMESTAMP masks

The following chart shows you the syntax for the PLAN, PACKAGE, and TIMESTAMP masks:

Keyword	Operator	Operand
PLAN/PACKAGE	>	planname/packageName
PLAN/PACKAGE	<	planname/packageName
PLAN/PACKAGE	>=	planname/packageName
PLAN/PACKAGE	<=	planname/packageName
PLAN/PACKAGE	<> (not equal)	planname/packageName
PLAN/PACKAGE	LIKE	planmask/packagemask Allows wildcards % and _
PLAN/PACKAGE	BETWEEN	planname1/packageName1 and planname2/packageName2
TIMESTAMP		YYYY-MM-DD-HH.MM.SS

Chapter 19. Running the Extract

Introduction

This chapter tells you how to plan for and execute your extract. It provides examples of normal extract and purge runs, as well as runs to restart a job that terminated abnormally, to convert from a previous version of !DB/EXPLAIN to the current release, and to estimate disk space required to hold your extract. It also discusses how you can obtain and display extract history.

Chapter contents

Planning for Extract Runs	352
Executing the Extract	354
Purpose of Your Extract Run	359
Function Options for Normal Extract Runs	361
Function Options for Normal Purge Runs	366
Normal Extract and Purge Summary	367
Input Syntax	370
Extract Input Examples	373
Purge Input Examples	375
Other Types of Extract Runs	376
Debugging Your Extract	378
Using Extract History	380

Planning for Extract Runs

Overview

This unit poses some questions you should ask yourself when planning to perform an extract and some actions to take prior to performing the extract.

Questions to answer

When planning for your extracts, you need to make the following decisions:

- How often will periodic extracts be run?
- Will you permit the running of ad-hoc extracts?
- What is the purpose of a given run (EXTRACT, PURGE)? See “Purpose of your Extract Run” on page 359.
- What required function option (REPLACE, MERGE, MERGE SINCE) will be used when running an extract? The effect of using one of these function options is discussed in “Specifying function options” on page 360.
- How often will you purge your VSAM clusters of extracted data?
- What type of data retention will be used? How many iterations of historical data will be retained?
- Will you permit the running of ad-hoc purges?
- What is your backup and recovery strategy? What considerations are specific to your installation?

The answers to these and other questions are installation-specific and will depend on your objectives for running !DB/EXPLAIN.

Actions to take

As an adjunct to planning for your extract runs, you should consider the following actions:

- Generate required JCL or customize the extract job (*db2idJXTE*) in the CNTL PDS as necessary to execute extracts and purges.

Note: During a refresh, the extract JCL is overlaid. If you want to save your customization, you should take appropriate actions to save the extract job in another PDS member.

- Build and schedule jobstreams for recurring events—periodic extracts and purges and VSAM cluster backups.
- Schedule jobs to prevent conflicts with online access.
- Execute ad hoc extracts and purges as required.
- Coordinate online access for ad hoc extracts and purges.

Restriction on concurrent extracts

You cannot run concurrent extracts to the same set of extract files.

Executing the Extract

Overview

This unit tells you how to prepare to execute the extract by reviewing and modifying procedure *db2idJXTE*.

Background about the batch extract job

Job *db2idJXTE* that performs the extract is delivered as part of the !DB/EXPLAIN installation job stream. It is placed in the !DB/EXPLAIN *hilev.CNTL* library as part of the installation process. During the installation process, you used the installation input panels to tailor some parameters. This tailoring is saved in the JCL. While you are using !DB/EXPLAIN, you can generate extract JCL by executing the JCL? command, selecting option 6, and modifying the JCL with any values specific to your installation. During JCL generation, you can also specify whether to modify or add any SYSIN control statements required to execute the extract.

Caution when running the extract job

If you run batch extracts using !DB/EXPLAIN Version 235 or later and are using existing production JCL from a version prior to Version 235, you must modify the DCB LRECL attributes for the DBRM and PACK DD statements. Refer to the *db2idJXTE* job for acceptable values. Failure to modify these values will result in an error message and your job will abend.

Tasks to complete to prepare for execution

You can generate batch JCL using the JCL? command. You can also review the example of the batch extract JCL contained in member *db2idJXTE* in your *hilev.CNTL* library. Your actual JCL will reflect the tailoring you have performed during installation. Ensure that the tasks listed in this table are complete before running the batch extract.

Task you want to perform	Information source
To understand its functionality, review the JCL in member <i>db2idJXTE</i> in <i>hilev.CNTL</i> or the JCL you created using the JCL command.	The generated JCL or the model job stream
Ensure the syntax of the input job stream is correct.	“Guidelines for extract JCL syntax” on page 356.
Ensure any additional customization you need to perform has been accomplished.	“Customizing the extract JCL member” on page 357.
Specify the purpose of the extract run using positional parameter 1 of the EXTRACT statement in the job.	“Purpose of your Extract Run” on page 359.
Include the SYSIN control statements to specify additional parameters and extract function option(s) for your extract run.	Example statements in “Extract Input Examples” on page 373.
Specify whether or not this is a debug run and if so, what type of debug run, using positional parameters 2 through 4 of the EXTRACT statement in the job.	“Debugging Your Extract” on page 378.
Estimate DASD for VSAM clusters and non-VSAM datasets.	“Estimating Disk Storage Required by an Extract” on page 382.

Guidelines for extract JCL syntax

- Enclose strings in quotes (“ ”) or in apostrophes (‘ ’).
- If an apostrophe or a quote appears in the character string, that nested delimiter must be repeated. For example, to display the contraction can't, the correct syntax is ‘can’’t’.
- Text can be either upper or lower case. All text that is not contained between quotes or apostrophes is rendered upper case.
- Any line having an asterisk (*) in column 1 is treated as a comment except for a continuation line used to continue a string.
- Text must lie between columns 1 and 71 inclusive. There must be at least one blank between words; more than one blank is tolerated. To continue a string to the next line, place any non-blank character in column 72. The contents of column 72 are not used. Placing a non-blank character in column 72 permits you to split a word or a string enclosed in quotes or apostrophes across several lines. The continued string starts in column 1 of the next line.
- Fields, except for masks, that do not contain data to their specified length are padded on the right with blanks. Masks are not padded with blanks.
- To include significant blanks in a PACKAGEMASK and/or a PLANMASK, contain the string in either quotes or apostrophes. Note that PLAN LIKE %X% is not the same as PLAN LIKE ‘%X%bbb’. A target matches the first example when X appears anywhere in the target. A target matches the second example when X appears anywhere in the first positions of the target and three blanks appear at the end of the target.
- Quotes or apostrophes are necessary to accommodate any of the following characters inside a string:

"	quote
'	apostrophe
(blank)	blank
<	less than
>	greater than
=	equal
¬	not sign
(left parenthesis
)	right parenthesis

They are also required in order to suppress translation (folding) to upper case where the target column data is in lower case.

Guidelines for extract JCL syntax (continued)

- Comparison operators such as less than (<) or equal to (=) can be adjacent to the values they reference, or there may be interposed spaces. Thus the following phrases are identical:

PLAN > B

PLAN> B

PLAN >B

PLAN>B

Customizing the extract JCL member

You should review the extract JCL to customize it as follows:

- For restartability, change your non-VSAM datasets to permanent files (see “Specifying permanent storage and work datasets (optional)” on page 397).
- Review the allocation for SYSOUT and SORTWK to ensure adequate space has been allocated. The space allocated to the sort work files depends upon the volume of data being sorted. Sort messages in SYSOUT can indicate a space shortage. Review these messages and increase workfile space as the volume of data being extracted increases.
- Review the allocation for SYSPRINT, SYSTEM, STDERR, and SYSUDUMP. You should monitor the output regularly for important error and informational messages.

Overriding dynamic dataset allocation

You can override dynamic dataset allocation that occurs as part of the installation process by including appropriate DD statements in the extract JCL. DD statements should be of the form **//ddname DD DSN=appropriate.dsn,DISP=SHR.**

The extract program dynamically allocates:	Override dynamic allocation by:
<p>SYSCATLG, SYSSTMT, and SYSPLAN with disposition SHR to the cluster dataset names retrieved from the extract ID profile entry.</p> <p>Note: Even though these datasets are allocated with DISP=SHR, you can avoid concurrent online and batch access by setting the Enqueue on Extract datasets field on the Tuning Parameters panel to Y (the default). See “Reorganization requires exclusive access” on page 390.</p>	<p>Including the ddnames SYSCATLG, SYSSTMT, and SYSPLAN in the <i>db2idJXTE</i> job stream.</p>
<p>!DB/EXPLAIN load library to the ddname CANDLLIB for dynamic module loads after the dataset name for the !DB/EXPLAIN load library is extracted from the appropriate profile entry.</p>	<p>Including the ddname CANDLLIB in the <i>db2idJXTE</i> job stream.</p>
<p>!DB/EXPLAIN message library to the ddname KTEMSG after the dataset name for the !DB/EXPLAIN messages is extracted from the appropriate profile entry.</p>	<p>Including the ddname KTEMSG in the <i>db2idJXTE</i> job stream.</p>
<p>The ddname DSNLIB to the DB2 DSNLOAD that matches the DB2 version specified in the profile for the executing extract ID. (The extract requires dynamic loading of interface modules.)</p>	<p>Including the ddname DSNLIB <i>ddname</i> in the <i>db2idJXTE</i> job stream.</p>

Purpose of Your Extract Run

Overview

This unit describes the various purposes an extract run can have and refers you to appropriate units of this book for details on specifying extract function options.

Purpose for running an extract

You can run an extract to extract data from the DB2 catalog or plan table, to purge data from VSAM files, to handle error conditions, and to determine extract storage requirements. The run purpose and whether or not the run is a debug run are specified on the EXTRACT dd statement of the extract job.

You must specify extract function options for normal runs whose purpose is either extract or purge. Your extract input specifies the extract function option(s) you want by providing the SYSIN control statements in the extract JCL.

Specifying the run purpose

The extract run purpose is controlled by the first positional parameter of the extract program KTEXTRAC in the *db2idJXTE* job. The table below shows the purposes for running an extract.

Extract Purpose	Extract Action
Normal Run (Extract)	Extracts Catalog data or PLAN_TABLE data and places the data into VSAM files (see “Function Options for Normal Extract Runs” on page 361).
Normal Run (Purge)	Purges data from VSAM files (see “Function Options for Normal Purge Runs” on page 366).
Restart Run	Recovers from a failed extract that was using internal checkpointing (see “Restart runs” on page 376).
Force Run	Overrides a restart (see “Force runs” on page 376).
Dummy Run	Performs a space calculation but does not perform any VSAM input/output processing (see “Dummy runs” on page 377).
Convert Run	Causes an extract conversion utility to be run (see “Convert runs” on page 377).

Specifying function options

To specify function options, provide the SYSIN control statements containing your extract input for the KTEXTRAC program in the *db2idJXTE* JCL. See the following units for detailed information:

Information You Want	Information Source
Description of function options for normal extract and purge runs	“Function Options for Normal Extract Runs” on page 361 “Function Options for Normal Purge Runs” on page 366
Extract and purge summary table showing required and optional input	“Summary table” on page 367
Syntax diagrams of the SYSIN control statements in procedure <i>db2idJXTE</i>	“Input Syntax” on page 370
Examples of the SYSIN control statements to execute various function options of the extract and purge runs	“Extract Input Examples” on page 373

Function Options for Normal Extract Runs

Overview

This unit describes the extract function options for a normal extract run.

Required function option keywords

A normal extract run extracts data from the DB2 catalog or plan table. You are not required to specify a value for positional parameter 1 of the extract program but you must specify one of the following required extract function options in the SYSIN input:

Extract Function Option	Function Option Action
REPLACE	<p>Deletes all records in the extract ID before extracting new data from the DB2 catalog. REPLACE is like DISP=OLD.</p> <p>If you plan to save and view historical extracts, <i>do not</i> use REPLACE. Using REPLACE causes the current extract to wipe out any historical extract data you might have saved.</p>
MERGE	Accumulates data into the extract ID, allowing for the collection of history.

Required function option keywords (continued)

Extract Function Option	Function Option Action
MERGE SINCE	<p>Valid for Catalog extracts only, MERGE SINCE is a form of MERGE that extracts data bound since the last MERGE or MERGE SINCE extract. MERGE SINCE is an incremental extract that requires a previous REPLACE or MERGE extract to have been run using the identical extract ID and plan or package specification masking.</p> <p>MERGE SINCE has two optional parameters:</p> <ul style="list-style-type: none"> ● MERGE SINCE TIMESTAMP <p>Extracts data that was bound since a time specified by you in the TIMESTAMP parameter you enter. This incremental extract differs from MERGE SINCE as follows:</p> <ul style="list-style-type: none"> – You can use this function for an extract ID that has no existing data (that is, you can run MERGE SINCE TIMESTAMP instead of REPLACE). – It is not necessary to have run a previous extract that used identical masking (see “Extract Masking” on page 348). ● DROP <ul style="list-style-type: none"> – Flags all dropped objects since the last MERGE SINCE <p>Thus, all plans and packages that were freed, bound, or rebound are flagged as historical in the extract dataset.</p> – Defaults to OFF <p>Note: For MERGE SINCE runs, we recommend you define your work files to be permanent datasets to preclude the potential loss of data in the event of a failure. You can delete these datasets upon successful completion of the job.</p>

Optional function option keywords

On a normal extract run, in addition to the required function options specified in the previous section, you can include these optional function options in your SYSIN input:

Extract Function Option	Function Option Action
THRESHOLD	A value that specifies the number of VSAM records that can be inserted by an extract before !DB/EXPLAIN triggers an automatic reorganization of your VSAM clusters. To activate automatic reorganization, your VSAM clusters must be allocated with REUSE. THRESHOLD applies to the SYSCATLG and SYSSTMTS VSAM clusters only.

Optional function option keywords (continued)

Extract Function Option	Function Option Action
ALIAS	<p>Determines which of three methods will be used to research DB2 aliases:</p> <ul style="list-style-type: none"> ● None Specifying NO suppresses alias research. Specifying NO has the least impact on extract resource utilization. Useful if your installation does not have a significant number of aliases. ● DB2 inquiry Specifying ASK_DB2 executes a DB2 inquiry to determine whether the qualified name is an alias each time a table is used. This method maximizes processing but minimizes the use of virtual memory. If your installation uses a large number of aliases, and if virtual storage or working set sizes are critical, and if alias research must be performed, you may find this method to be the preferred one. ● Virtual Storage table lookup (default) Specifying IN_MEMORY causes the extract to read aliases into a virtual memory table one time and check the table using the qualified name each time a table is used. This method minimizes processing in exchange for virtual storage utilization. If an installation does not use an extremely large number of aliases and has no virtual storage or working set size problems, this is by far the most efficient processing method. <p>Note: If you specify NO here, an online execution of !DB/EXPLAIN displays the table's alias on the Tables panel. Specifying ASK_DB2 or IN_MEMORY causes !DB/EXPLAIN to display the table's actual name on the Tables panel during an online execution.</p>

Optional function option keywords (continued)

Extract Function Option	Function Option Action
SYNONYM	<p>Determines which of two methods will be used to research synonyms:</p> <ul style="list-style-type: none"> ● DB2 inquiry Specifying ASK_DB2 executes a DB2 inquiry to determine whether the qualified name is a synonym each time a table is used. This method maximizes processing but minimizes the use of virtual memory. If your installation uses a large number of synonyms, and if virtual storage or working set sizes are critical, and if synonym research must be performed, you may find this method to be the preferred one. ● Virtual Storage table lookup (default) Specifying IN_MEMORY causes the extract to read synonyms into a virtual memory table one time and check the table using the qualified name each time a table is used. This method minimizes processing in exchange for virtual storage utilization. If an installation does not use an extremely large number of synonyms and has no virtual storage or working set size problems, this is by far the most efficient processing method.

Function Options for Normal Purge Runs

Overview

This unit describes the extract function options for a normal purge run.

Required function option keywords

A purge run erases data from the VSAM clusters. You are not required to specify a value for positional parameter 1 of the extract program but you must specify one of the following required extract function options in the SYSIN input:

Purge Function Option	Function Option Action
GENERATION	Specifies how many historical generations of data to keep and purges any remaining.
TIMESTAMP	Causes data with a bind date/time prior to the value specified to be purged. Purge processing requires that the format of TIMESTAMP be, at a minimum, YYYY-MM-DD (year, month, day). You may also optionally include time of day in the format HH:MM:SS (hours, minutes, seconds).

Optional function option keywords

On a normal purge run, in addition to the required function options specified in the previous section, you can include this function option in your SYSIN input.

Purge Function Option	Function Option Action
SYNC	Ensures that any records in the PLAN_TABLE file has parents in the Catalog VSAM file. If a record has no such parents, SYNC processing deletes that record from the PLAN_TABLE to ensure that the Catalog and PLAN_TABLE are synchronized.

Normal Extract and Purge Summary

Overview

The extract summary table in this unit identifies the run purpose, description of the run, required and optional input, and extract function options for normal extract and purge runs.

Summary table

Table 2 summarizes the required input for normal extract and purge runs.

Table 2 (Page 1 of 3). Extract Run Purpose—Extract or Purge				
Purpose	Description	Required Input	Optional Input	Extract Function Options
Extract PLAN_TABLE	Select data from a PLAN_TABLE.	Extract ID, plan_table owner, catalog owner, and one extract function option	Plan or package specification masking	Replace Replace the data within specified PLAN_TABLE VSAM cluster extract ID with data selected for this run. Merge Merge the data selected for this run with data already in specified PLAN_TABLE VSAM cluster extract ID.

Table 2 (Page 2 of 3). Extract Run Purpose—Extract or Purge				
Purpose	Description	Required Input	Optional Input	Extract Function Options
Extract Catalog	Select catalog data and statements pertaining to plans/packages.	Extract ID, catalog owner, and one extract function option	Plan or package specification masking, threshold value, drop, timestamp, and alias.	<p>Replace Replace the data within specified catalog and statement VSAM cluster extract ID with data selected for this run. <i>Do not</i> use REPLACE if you plan to save and view historical extracts as REPLACE wipes out any historical extract data.</p> <p>Merge Merge the data selected for this run with data already in specified catalog and statement VSAM cluster extract ID.</p> <p>Merge Since Merge historically by bind date/time the data selected for this run with data already in specified catalog and statement VSAM cluster extract ID. Note: Since the purpose of this function is to select those plans or packages bound since the last extract, a previous REPLACE or MERGE extract <i>must</i> have been executed using the identical extract ID and plan/package specification masking as used with this function.</p> <p>Merge Since Timestamp Merge historically all data with a bind date/time after the timestamp value entered as a parameter. The data selected for this run is merged with data already in specified catalog and statement VSAM cluster extract ID. Note: It is not necessary to have run a previous extract that used identical masking. It is also possible to use this extract function option for an extract ID that has no existing data in it.</p>

Table 2 (Page 3 of 3). Extract Run Purpose—Extract or Purge				
Purpose	Description	Required Input	Optional Input	Extract Function Options
Purge Catalog/ Purge PLAN_ TABLE/ Purge Both	Purge data from VSAM clusters.	Extract ID and VSAM file type where file type is PLAN_ TABLE, catalog (including the VSAM statement cluster), or both, and one extract function option.	Plan or package specification masking, SYNC option	<p>Generation Specifies how many historical generations of data to keep and purges any remaining.</p> <p>Timestamp Causes data with a bind date/time prior to the value specified in this parameter to be purged.</p>

Input Syntax

Overview

This unit provides the input syntax for normal extracts and purges of the DB2 catalog or plan table.

JCL to which the syntax applies

To understand the syntax diagrams in context, you should review the batch extract JCL you have created or that which is distributed with !DB/EXPLAIN.

EXEC statement portion of the JCL

This example illustrates the EXEC statement portion of the JCL.

```

/*=====
/*
/* PROC NAME:  KTEEXTRC
/*-----
/* FUNCTION:   EXECUTE !DB/EXPLAIN BATCH EXTRACT
/*-----
:
000015 /*=====
000016 //EXTRACT EXEC PGM=KTEEXTRAC,REGION=5120K, 1
000017 //      PARM='XXXXXXXX X X X X'
000018 /*
000019 /*      |           |           Debug parameters here
000020 /*      |           |           Restart parameter here
000021 /*=====

```

SYSIN portion of the JCL

This example illustrates the SYSIN portion of the JCL.

```

/*
000098 /* DB/EXPLAIN Extract Commands.
000099 /*
000100 //SYSIN DD * 2
000101 EXTRACT CATALOG
000102 EXTRACT_ID DB31
000103 COWNER SYSIBM
000104 REPLACE

```

Syntax diagram: extract from catalog

To cause a normal extract from the DB2 catalog, your EXEC statement (**1**) should be of the form **EXTRACT EXEC PGM=KTEXTRAC,PARM=XXXXXXXX** where **XXXXXXXX** can be any value except **RESTART**, **FORCE**, **DUMMY**, or **CONVERT** and the SYSIN input (**2**) has the following syntax:

```

>> ---EXTRACT --CATALOG --EXTRACT_ID  extractid ----->
> ----COWNER   catalog owner ----->
>----REPLACE ----->
>  |--MERGE -----|
>    |--SINCE-----|
>      |--DROP---| |-- TIMESTAMP  timestamp --|
>----->
>  |--THRESHOLD threshold-value ----|
>----->
>  |--ALIAS ---IN_MEMORY -----| |SYNONYM ---IN_MEMORY--|
>    |--NO ---|                    |--ASK_DB2-|
>    |--ASK_DB2 -----|
>-----<<
>  |--plan specification --| |--package-specification --|

```

Syntax diagram: extract from plan table

To cause a normal extract from the DB2 plan table, your EXEC statement (**1**) should be of the form **EXTRACT EXEC PGM=KTEXTRAC,PARM=XXXXXXXX** where **XXXXXXXX** can be any value except **RESTART**, **FORCE**, **DUMMY**, or **CONVERT** and the SYSIN input (**2**) has the following syntax:

```

>>---- EXTRACT -- PLANTABLE -- EXTRACT_ID  extractid ----->
>---- POWNER   plan_table owner -- COWNER   catalog owner -->
>--- REPLACE ----->
>  |-- MERGE -----|
>-----<<
>  |-- plan-specification --| |-- package-specification--|

```

Syntax diagram: purge

To purge data from the catalog, plan table, or both, your EXEC statement (**1**) should be of the form **EXTRACT EXEC PGM=KTEXTRAC,PARM=XXXXXXXX** where **XXXXXXXX** can be any value except **RESTART**, **FORCE**, **DUMMY**, or **CONVERT** and the SYSIN input (**2**) has the following syntax:

```
>>---- PURGE ----- CATALOG ----- EXTRACT_ID  extractid---->
          |-- PLANTABLE --|
          |-- BOTH      --|

>--- GENERATION  generation  --|----->
  |-- TIMESTAMP  timestamp   --|      |-- SYNC --|

>-----<<
  |-- plan-specification --| |-- package-specification--|
```

Extract Input Examples

Overview

The examples in this unit show you how to code your extract function options that provide the SYSIN control statements in job *db2idJXTE*.

Examples

To provide the SYSIN control statements, follow the examples in the table below:

To execute...	Provide SYSIN Control Statements...
Extract—Replace	<p>EXTRACT CATALOG EXTRACT_ID <i>eeeeeeee</i> COWNER <i>ccccccc</i> REPLACE</p> <p>Where:</p> <ul style="list-style-type: none"> ● <i>eeeeeeee</i> is the extract ID ● <i>ccccccc</i> is the Catalog owner
Extract—Merge	<p>EXTRACT CATALOG EXTRACT_ID <i>eeeeeeee</i> COWNER <i>ccccccc</i> MERGE</p> <p>Where:</p> <ul style="list-style-type: none"> ● <i>eeeeeeee</i> is the extract ID ● <i>ccccccc</i> is the Catalog owner
Extract—Merge Since	<p>EXTRACT CATALOG EXTRACT_ID <i>eeeeeeee</i> COWNER <i>ccccccc</i> MERGE SINCE PLAN < <i>x</i></p> <p>Where:</p> <ul style="list-style-type: none"> ● <i>eeeeeeee</i> is the extract ID ● <i>ccccccc</i> is the Catalog owner ● <i>x</i> is the plan mask. <p>The Merge Since extract <i>must</i> match an original (REPLACE) extract.</p>

Examples (continued)

To execute...	Provide SYSIN Control Statements...
Extract—Merge Since Timestamp	<pre>EXTRACT CATALOG EXTRACT_ID eeeeeeee COWNER cccccccc MERGE SINCE TIMESTAMP timestamp</pre> <p>Where:</p> <ul style="list-style-type: none"> ● eeeeeeee is the extract ID ● cccccccc is the Catalog owner ● timestamp is the timestamp. All plans and packages bound after this timestamp are extracted. Those bound before this timestamp are not extracted. <p>The Merge Since Timestamp extract <i>does not</i> have to match an original (REPLACE) extract.</p>
Extract (from a user-defined plan table)	<pre>EXTRACT PLANTABLE EXTRACT_ID eeeeeeee POWNER pppppppp COWNER cccccccc REPLACE</pre> <p>Where:</p> <ul style="list-style-type: none"> ● eeeeeeee is the extract ID ● pppppppp is the PLAN_TABLE owner ● cccccccc is the Catalog owner

Notes:

1. Any extract types can have any plan or package mask used with the operators =, >, <, >=, <=, or **BETWEEN**. For example:
 - PACKAGE = packagename
 - PLAN > planname
 - PACKAGE BETWEEN packagename AND packagename.
2. If you are migrating from !DB/EXPLAIN Version 220 or earlier to !DB/EXPLAIN Version 230 or later, you cannot carry your historical plan table data forward. However, in Version 230 or later, you *can* retrieve this history by identifying the PLAN_TABLE from the earlier release as a user-defined plan table to bring it forward.
3. To run an extract from a PLAN_TABLE, you must have previously performed a catalog extract on a DB2 subsystem having the same plans.
4. You can merge PLAN_TABLE data from different users into the same extract ID by using the optional MERGE keyword. To do so, run the EXTRACT PLANTABLE job one time for each PLAN_TABLE owner, specifying the same extract ID for each job, and specifying the MERGE function option.

Purge Input Examples

Overview

The examples in this unit show you how to code your purge function options that provide the SYSIN control statements in job *db2idJXTE*.

Examples

To provide the SYSIN control statements, follow the examples in the table below:

To execute...	Provide SYSIN Control Statements...
Purge (historical data)	<p>PURGE CATALOG EXTRACT_ID <i>eeeeeeee</i> GENERATION <i>generation</i></p> <p>Where:</p> <ul style="list-style-type: none"> ● <i>eeeeeeee</i> is the extract ID ● <i>generation</i> is the number of generations to save. All generations previous to those specified here (higher, numerically) are purged.
Purge (from a given bind date)	<p>PURGE CATALOG EXTRACT_ID <i>eeeeeeee</i> TIMESTAMP <i>timestamp</i></p> <p>Where:</p> <ul style="list-style-type: none"> ● <i>eeeeeeee</i> is the extract ID ● <i>timestamp</i> is the bind date timestamp value you specify. All records extracted prior to that date are purged.

Other Types of Extract Runs

Overview

This unit provides information about types of extract runs other than extract runs or purge runs.

Restart runs

To cause a restart run, positional parameter 1 of the extract program KTEXTRAC must be set to RESTART.

- A restart run must follow an extract run that has failed to complete successfully. Use the same extract input as the failed run, and set positional parameter 1 to RESTART.
- A restart run automatically picks up the last checkpoint location and recovers/synchronizes VSAM clusters without requiring a restore (if the VSAM clusters were not damaged).
- A restart run requires the files to be in the state they were in when the previous run failed. Damaged files prevent the restart from executing.

Note: In order to execute a restart, the extract work files used in the original run must have been defined as permanent datasets. The extract work datasets can be deleted at the end of the job if the return code = 0. (See “Specifying permanent storage and work datasets (optional)” on page 397 for information on the datasets that must be made permanent.)

Force runs

To cause a force run, positional parameter 1 of the extract program KTEXTRAC must be set to FORCE. A force run overrides the restart.

A force run is a normal run with any variation of extract input and the force parameter set to FORCE. If an extract run has failed to complete successfully, specifying FORCE allows a normal extract run to execute without first running a restart. However, a force run will not correct any logical file errors.

FORCE should only be executed as part of a Catalog REPLACE run. If used with MERGE or MERGE SINCE, your data can be corrupted. Because FORCE should only be used with Catalog REPLACE, use of the FORCE parameter causes your extract history to be overwritten and therefore lost.

To ensure data integrity, you must either run the extract using RESTART, or restart the extract from system backup files.

Dummy runs

To cause a dummy run, positional parameter 1 of the extract program KTEXTRAC must be set to DUMMY. A dummy run causes a space calculation to be performed, but does not cause any VSAM input/output processing. Work datasets hold intermediate results required for the space calculation.

Convert runs

Convert runs are performed to upgrade from a previous version or maintenance level of !DB/EXPLAIN to the current version or maintenance level. To cause a convert run, positional parameter 1 of the extract program KTEXTRAC must be set to CONVERT. A convert run causes an extract conversion utility to be run to provide statement consistency, to reorganize the SYSSTMT VSAM cluster, and to enhance performance.

To run a CONVERT, use a copy of your current production JCL and perform the following steps:

Step	Action
1	Change the STEPLIB to point to the new load library containing the new KTEXTRAC load module.
2	Change the <i>first</i> parameter on the EXEC statement to CONVERT .
3	Use any valid SYSIN DD statement that contains the extract ID for the SYSSTMT VSAM cluster. You can use an EXTRACT or PURGE SYSIN DD statement; the conversion ignores these values and the result is the same. Note: You must run a CONVERT for each SYSSTMT VSAM cluster you have defined.

Caution: If you run batch extracts using !DB/EXPLAIN Version 235 or later and your existing production JCL is from a version prior to Version 235, you must modify the DCB LRECL attributes for the DBRM and PACK DD statements. Refer to the *db2idJXTE* job for acceptable values. Failure to modify these values will result in an error message and your job will abend.

Debugging Your Extract

Overview

This unit tells you the types of debugging available to you and how to activate each type.

Types of debugging runs

Use positional parameters 2 through 4 of the KTEXTRAC program to specify whether or not a particular extract run is for debugging purposes, and if so, what type of debugging you want. Four types of debugging are possible:

Structure Dumps internal structures

I/O Traces I/O activity

Flow trace Traces program flow


Parser trace Traces parser activity

Note: Because debugging is very resource intensive, it should be performed only when requested by Candle Support Services or Research and Development. Debugging is available on all extract types. Debug messages are printed to SYSPRINT.

Parameters for specifying debugging

The KTEXTRAC dd statement is of the form:

```
EXTRACT EXEC PGM=KTEXTRAC, PARM='XXXXXXXX X X X X'.
```



Parameters 2 through 5

The table below supplies the values to be placed in positional parameters 2 through 5 of KTEXTRAC to perform debugging.

Positional Parameter Number	Value
2—structure debug switch	D Activate structure debugging any other value or blank Inactivate structure debugging
3—I/O debug switch	D Activate I/O debugging any other value or blank Inactivate I/O debugging
4—flow trace debug switch	D Activate flow tracing any other value or blank Inactivate flow tracing
5—parser trace switch	D Activate parser tracing any other value or blank Inactivate parser tracing

Using Extract History

Overview

This unit tells you how to view, print and save historical extracts.

Specifying the number of generations to display

You can specify the number of generations of extract history !DB/EXPLAIN displays or prints in the Generations of Catalog History field on the Housekeeping panel Extract Processing Defaults.

Saving historical extracts

If you want to save historical extracts, do not use the function option REPLACE. This function option deletes any previous extract histories making them unavailable for viewing or printing.

Viewing extract history

To view extract history online, follow the steps in the table below:

Step	Action
1	From the !DB/EXPLAIN Primary Menu, select Extract History.
2	Press Enter. Result: !DB/EXPLAIN displays the Extract History panel. This panel lists historical information for the three extract datasets: DB2 catalog, PLAN_TABLE, and statements.

Printing extract history

To print extract history as a !DB/EXPLAIN batch job, include the following commands in your SYSIN input as part of the !DB/EXPLAIN batch utility *DB2idUTIL*:

```

HI      (object class = extract history)
PRNT   (send output to REPORTS dd)
END

```

Chapter 20. Disk Storage for the Extract

Introduction

This chapter describes the techniques you can use to plan for and estimate the storage required to hold your !DB/EXPLAIN extract. It also discusses considerations related to reorganization and recovery.

Chapter contents

Estimating Disk Storage Required by an Extract	382
Estimating Space for Non-VSAM Files	386
Planning for VSAM Clusters	388
VSAM Fragmentation and REORG Considerations	390
Manual Reorganization	392
Automatic Reorganization	394
Recovering Your VSAM Clusters	395
Restarting after a Failure	396

Estimating Disk Storage Required by an Extract

Overview

This unit provides information on estimating the disk storage required by an extract. The disk storage can contain both VSAM and non-VSAM files. You will also learn other factors you should consider when estimating storage.

Preparatory steps

Before running the extract, you should perform the following steps to determine the amount of disk storage you require to run an !DB/EXPLAIN extract.

Step	Action
1	Use the information in this unit to estimate the disk space required for VSAM and non-VSAM files.
2	Calculate the amount of DB2 data to be extracted. To do so, you may need to select your data—plans, packages, and statements—using the DUMMY parameter on an extract run.
3	Total the data to be extracted for all extract IDs.
4	Use the calculated data total together with the values and suggestions in this unit to revise and improve your disk space estimates.

Background about space estimates

Space estimates in the space calculations in this section are specified in bytes. You may need to convert these to the appropriate unit for the device type(s) used in your location, for example, CYL, TRK, MEGABYTE, or KILOBYTE.

Estimating space for the VSAM clusters

The table below provides values in bytes that you can use to estimate space for your VSAM clusters. Once you have made an initial calculation, modify it by assessing the impact of the factors contained in the unit “Other factors you can use to refine estimates” on page 384.

Cluster Name and contents	Estimated Space in Bytes
SYSPLAN: PLAN_TABLE Cluster	<ul style="list-style-type: none"> ● 250 bytes per extracted PLAN_TABLE entry ● 1800 bytes per statement EXPLAINED during an !DB/EXPLAIN run
SYSCATLG: Catalog Cluster	<ul style="list-style-type: none"> ● 1750 bytes per plan or package as an initial estimate <p>Note: The number of plans and packages depends upon how the extract ID is configured, the type of masking used for each extract ID, and how many historical records are extracted if the MERGE SINCE option is used.</p>
SYSSTMT: Statements Cluster	<ul style="list-style-type: none"> ● 279 bytes per plan statement ● 289 bytes per package statement

Other factors you can use to refine estimates

Apply the factors below to your calculated estimate for VSAM clusters to refine the estimate:

- To obtain an automatic calculation of the VSAM data bytes that would be created by a given extract, run the extract with PARM=DUMMY.
- Use approximate space values initially. You can resize the file later if necessary by running KTEREBLD. See “Performing a manual reorganization or resizing your extract files” on page 393 for information on this job.
- It is preferable to estimate large secondary file extents or overestimate the primary file extents instead of abending.
- The size of a given plan, package, or DBRM varies widely from any other plan, package, or DBRM. Try to determine the average size for your installation to use as a tuning factor. You may want to query the DB2 catalog to obtain approximate sizes.
- The number of plan or package statements in an extract varies widely from extract ID to extract ID. Again, try to determine an average to be used for your installation.
- The number of and use of a group of extract IDs per VSAM cluster varies widely from installation to installation.
- The number of plan table records directly relates to the amount of !DB/EXPLAIN activity that is performed. You can approximate the size of the PLAN_TABLE using the following formula:

$$\begin{array}{r}
 \text{\# of EXPLAINable statements contained in the extract file} \\
 \times \\
 \text{1800 bytes per statement} \\
 \times \\
 \text{\# of generations of history retained} \\
 + \\
 \text{an appropriate blocking factor}
 \end{array}$$

- Some types of !DB/EXPLAIN records are not carried forward from release to release. Other record types change in size. When moving to a new !DB/EXPLAIN release, you should review your estimates to determine whether release-dependent changes must be made.

Obtaining more information on VSAM utilities

If you want to learn more about the utilities that support the creation and maintenance of VSAM clusters, refer to the following IBM publications:

- IBM MVS/Data Facility Product: Using datasets, SC26-4749
- IBM MVS/Data Facility Product: Access Method Services Reference for ICF Catalogs, SC26-4500

Estimating Space for Non-VSAM Files

Overview

This unit provides the information necessary for estimating the DASD space required for each object, preprocess, and REORG file.

Planning for space for non-VSAM files

These non-VSAM files are needed until an extract run is complete; therefore, you must plan for the space to hold one complete run (either a successful run or an aborted run with a restart).

When estimating space, the number of bytes required for each file is determined by multiplying the file's logical record length by the number of records expected to be extracted.

To preclude abending, you should estimate large secondary file extents or overestimate the primary file extents. If you have insufficient DASD and you expect a large number of extract records, you can use multi-volume tape files.

!DB/EXPLAIN object files

Object Type	File Size and Format
DBRM —DBRM file	<ul style="list-style-type: none"> ● LRECL=240 ● BLKSIZE=6000 ● RECFM=FB
STMT —Plan statement file	<ul style="list-style-type: none"> ● LRECL=557 ● BLKSIZE=6127 ● RECFM=FB
PACK —Package statement file	<ul style="list-style-type: none"> ● LRECL=270 ● BLKSIZE=6210 ● RECFM=FB

Preprocess file

Preprocess files contain new records produced by the extract that must be merged into the VSAM clusters. When estimating space for preprocess files, use the following guidelines:

Preprocess Record Type	File Size and Format
OUTSTMT —New statement cluster records	<ul style="list-style-type: none"> ● Space estimate in bytes is a sum of the PACK and STMT file space estimates ● LRECL=797 ● BLKSIZE=12127 ● RECFM=VB
OUTCATLG —New catalog cluster records	<ul style="list-style-type: none"> ● Space estimate is 2500 bytes per plan or package extracted in the extract run ● LRECL=352 ● BLKSIZE=23476 ● RECFM=VB

REORG files

If automatic REORG is invoked as the result of a specified threshold value (see “VSAM Fragmentation and REORG Considerations” on page 390), then REORG files contain existing records from the VSAM clusters. When estimating space for REORG files, use the following guidelines:

REORG Record Type	File Size and Format
REOSTMT —Statements	<ul style="list-style-type: none"> ● Space estimate in bytes can be determined by executing an IDCAMS LISTCAT on the Statement VSAM cluster. ● LRECL=581 ● BLKSIZE=23476 ● RECFM=VB
REOCATLG —Catalog Records	<ul style="list-style-type: none"> ● Space estimate in bytes can be determined by executing an IDCAMS LISTCAT on the Catalog VSAM cluster. ● LRECL=352 ● BLKSIZE=23476 ● RECFM=VB

Planning for VSAM Clusters

Overview

This unit addresses factors you must consider when setting up your VSAM clusters to hold extract data.

Decisions you must make

The table that follows identifies some of the key planning decisions you must make when setting up your VSAM clusters. Where additional information is needed, you are referred to a unit containing additional details.

Step	Action
1	Determine the disk space required to hold your VSAM clusters. See “Estimating space for the VSAM clusters” on page 383 for directions on performing the space estimate.
2	Determine the number of datasets required to hold your extract data.
3	Establish naming conventions for your datasets.
4	<p>If multiple datasets are required to hold your extract data, determine how data will be assigned to those datasets. You should consider:</p> <ul style="list-style-type: none"> ● The processing needs of your installation ● Whether your installation uses one or many extract IDs and the implications of this decision on the distribution of data across clusters ● Your business organization and the information access needs of each organizational unit
5	Determine how you will separate data on physical volumes of disk storage. See “Placement of data on physical storage” on page 389 for factors affecting your decision.
6	Establish plans for recovery of your extract data in the event of a system failure. See “Recovering Your VSAM Clusters” on page 395 for additional information related to data recovery.

Placement of data on physical storage

When determining how best to physically organize your extract data on VSAM physical storage, you may choose to:

- Implement multiple sets of VSAM clusters to spread data across multiple DASD volumes
- Use key ranges to place data across multiple DASD volumes

Note: If you implement key ranges, you cannot use the VSAM REUSE option. This precludes your using the !DB/EXPLAIN extract's automatic reorganization facility.

Organization of extract data in VSAM physical storage

Within a VSAM key sequenced dataset (KSDS), your extract data is grouped:

- Within a VSAM cluster by extract ID.
- Within the extract ID by record type: plan, package, plan statement, package statement, PLAN_TABLE entry, table data, table space data, etc.
- Within record type by major field such as plan name, package name, collection ID, etc.
- Within a major field by time, such as BIND date/time, for historical purposes.

VSAM Fragmentation and REORG Considerations

Overview

When your VSAM control intervals and control areas become fragmented, you should consider reorganization. This unit provides some guidelines for reorganization and refers you to sources of additional information.

Determine percent of fragmentation

The nature of data collection and purging lends itself to fragmentation of VSAM control intervals (CI) and control areas (CA). A large number of CI and CA splits increases access time. To determine whether your VSAM storage suffers from excess fragmentation:

Step	Action
1	Regularly run VSAM LISTCATs to monitor VSAM cluster statistics.
2	Determine the percentage of CI and CA splits.
3	If the percentage of either CI or CA splits exceeds a value you select (Candle suggests 10% to 15%), reorganize the cluster.

Reorganization requires exclusive access

Whether you plan to do manual or automatic reorganization, the programs that perform the reorganization need exclusive access to the VSAM files. This means that a reorganization cannot be performed when !DB/EXPLAIN users are accessing the VSAM files online. To ensure that your online access to !DB/EXPLAIN and your performance of an extract with potential reorganization of your VSAM KSDSs are serial rather than concurrent, follow the steps in the table below:

Step	Action
1	On the !DB/EXPLAIN Housekeeping Menu, select the Tuning Parameters option. Result: The Tuning Parameters panel containing VSAM tuning parameters is displayed.
2	On the Tuning Parameters panel, ensure that the Enqueue on Extract datasets field is set to Y (the default).

Sources of additional information

See the following units for detailed information:

Information You Want	Information Source
Manual reorganization: Guidelines and steps to perform	“Manual Reorganization” on page 392
Automatic reorganization: Guidelines and steps to perform	“Automatic Reorganization” on page 394

Manual Reorganization

Overview

This unit describes when to do a manual reorganization and the procedure for doing so.

What files must be manually reorganized

While the VSAM catalog and statements (SYSCATLOG and SYSSTMTS) files can be manually reorganized, the preferred method is to automatically reorganize them according to the value you specify on the THRESHOLD parameter of an extract run (see “Automatic Reorganization” on page 394).

The PLAN_TABLE file cannot be reorganized automatically. Use the procedures in this unit to reorganize the PLAN_TABLE file.

PLAN_TABLE organization affects BUILD performance

It is important to put procedures in place to manually reorganize the PLAN_TABLE file according to criteria specific to your installation. The performance of the BUILD function is directly related to the degree of organization of the PLAN_TABLE file. A poorly organized PLAN_TABLE results in long BUILDS.

When do you use manual reorganization

Use manual reorganization:

- When you use key ranges to place your data across multiple DASD volumes. Use of key ranges precludes use of VSAM's REUSE option and, therefore, !DB/EXPLAIN's automatic REORG function.
- Your file size does not permit automatic REORG (such as when your extract is very large, or your total required space for the extract exceeds the space available).
- Your extract run elapse time exceeds that permissible for performing automatic REORG.
- Your site standards do not permit automatic REORG.

In those circumstances, you should plan to perform regular manual reorganizations.

Allocating a new set of !DB/EXPLAIN extract datasets

If you want to allocate a new set of !DB/EXPLAIN extract datasets, use the KTECLEAN batch utility. KTECLEAN performs a simple delete/define process. KTECLEAN is delivered as part of the !DB/EXPLAIN installation job stream and customized as part of the installation process, or you can generate appropriate KTECLEAN JCL using the JCL? command. Do not use KTECLEAN to perform a manual reorganization unless you want to discard your existing data. Instead, use KTEREBLD.

Performing a manual reorganization or resizing your extract files

Use KTEREBLD to resize or reorganize your extract files while saving the contents of the files. KTEREBLD is delivered as part of the !DB/EXPLAIN installation job stream or you can generate appropriate KTEREBLD JCL using the JCL? command. When using KTEREBLD, you can change volume and space definitions, but should not change record size attributes.

Building the JCL to run KTECLEAN or KTEREBLD

The easiest and fastest way to build the job to run KTECLEAN or KTEREBLD is to use the JCL? command. Follow these steps.

Step	Action
1	On the command line of any !DB/EXPLAIN panel, type JCL?
2	Press Enter. Result: !DB/EXPLAIN displays the JCL Generation panel.
3	On the JCL Generation panel, type 6 in the Options field.
4	Press Enter. Result: !DB/EXPLAIN displays the Extract JCL Generation panel.
5	Enter the appropriate value in the Option field of the Extract JCL Generation panel: <ul style="list-style-type: none"> ● To create the JCL for KTECLEAN, type 5 in the Option field. ● To create the JCL for KTEREBLD, type 6 in the Option Field.
6	Press Enter. Result: !DB/EXPLAIN displays a secondary panel on which you provide required and optional values. It then creates the JCL to run the job.

Automatic Reorganization

Overview

This unit describes the way in which an automatic reorganization is triggered and how you can control the trigger point.

How automatic reorganization is triggered

The extract uses the value you assign to THRESHOLD in the SYSIN input to determine when to perform an automatic reorganization. The value for THRESHOLD ranges from 0 through 99999999 with the default being 99999999. If the value for THRESHOLD is exceeded during an extract run, that is, the extract has inserted a number of VSAM records exceeding the value specified by THRESHOLD, the catalog and statements clusters are automatically reorganized after the extract run has completed.

Assigning a threshold value

The list below provides some factors to consider when you have decided to use the extract's automatic REORG facility and want to assign an appropriate THRESHOLD value:

- To determine the optimum value for THRESHOLD, review the VSAM file statistics in the extract output. Set THRESHOLD to 80% of the number of adds provided in the SYSCATLG statistics.

SYSCATLG statistics are displayed in message KTE00R8 of the extract output. An example follows:

```
KTE00R8 VSAM file statistics:  
File: SYSCATLG; READS:938; ADDS:120; DELETES: 120; UPDATES: 4
```

For this example, the number of added records is 120. Set the THRESHOLD value to 96.

- To force an automatic REORG, set the THRESHOLD value to 0 for any catalog extract.

Note: Data must be extracted to cause the REORG. If you set the THRESHOLD value to 0 and mask plans and packages so that no data is extracted, *no REORG is performed*.

Automatic REORG resource utilization is minimal and performance is favorable compared to IDCAMS.

Recovering Your VSAM Clusters

Overview

This unit tells how to plan for recovery of your VSAM clusters.

Planning for recovery

You should have procedures in place to back up your VSAM clusters at appropriate intervals. When planning for recovery, you will need to consider the following questions:

- Are system VSAM backups available?
- Is external VSAM activity logging available for recovery?
- Was REUSE specified in VSAM definitions?
- Are selected rows, preprocess, and REORG files available?
- Will you use system recovery rather than extract restart?
- Do your recovery plans consider factors related to file size and elapsed time to recover?

Restarting after a Failure

Overview

This unit identifies the parameters you must specify to ensure that you can restart your extract run after a failure.

Actions to take

- Use external sorts to create logical restart points. These restart points (or checkpoints) consist of history records written to the !DB/EXPLAIN extract datasets by the !DB/EXPLAIN extract.

If an extract run fails, you can:

- Rerun the extract using **RESTART** or **FORCE** unless the VSAM clusters were externally restored. (**FORCE** should be used cautiously. See “Force runs” on page 376 for some considerations regarding the use of a **FORCE** run.) Using **RESTART** or **FORCE** ensures that the catalog extract file and the statement extract file are brought back into synchronization.

On a **RESTART**, *do not* alter any definitions that were in effect during the failed run.

In order to perform a **RESTART**, you must have specified that the extract work files used in the original run be permanent datasets. These datasets can be deleted at the end of the job if the return code = 0. (See “Specifying permanent storage and work datasets (optional)” below.)

- Externally restore the VSAM clusters and rerun the extract as a normal extract run. Since the internal extract **RESTART** facility is not being used to recover the failed extract, you can use temporary or catalog datasets for the extract work files.
- To ensure that all non-VSAM files are available for a restart, specify them as **DISP=SHR**. They are then reused (overwritten) by the next extract unless the current extract failed to complete successfully.

Specifying permanent storage and work datasets (optional)

(Specify permanent storage and work datasets if you plan to use the RESTART parameter to restart in case of failure.)

The !DB/EXPLAIN extract JCL uses storage and work datasets to store extracted data before transferring it to the !DB/EXPLAIN extract datasets. These data sets are initially shipped as temporary datasets (**DSN=&&datasetname,DISP=(NEW,PASS)**). If you want to be able to restart the extract if it fails, you must specify permanent storage datasets; for example, (**DSN=MY.DATASET,DISP=(NEW,CATLG,CATLG)**).

Caution: When your job completes successfully, you must delete the work datasets you have catalogued before rerunning the job; otherwise you will receive a “duplicate dataset” message.

Step	Action
1	Specify the extract work datasets (DBRM, STMT, PACK)
2	Specify the storage datasets (OUTSTMT, OUTCATLG) for newly extracted data.
3	Specify the storage datasets (REOSTMT, REOCATLG) for previously extracted data.

Restarting after a Failure

Using Batch Facilities

Chapter 21. Using !DB/EXPLAIN in Batch

Introduction

!DB/EXPLAIN provides a batch facility that allows you to perform most of the !DB/EXPLAIN tasks in batch mode. Most of the filters, sorts, formats, selects, and commands that can be issued against an object during an online execution are supported in batch. When performing a large amount of work, Candle recommends you use the batch facility to conserve system resources.

This chapter describes the jobs that comprise the batch facility and general considerations for using !DB/EXPLAIN in batch mode. Two separate chapters document the major batch functions:

- Batch facilities for application development are described in “Batch Processing for Application Development” on page 419.
- Batch facilities for database administration and management are described in “Batch Processing for Database Administration” on page 439.

Chapter contents

Planning for Running Batch Jobs	402
Determining Which Batch Job to Use	403
Example JCL and Control Members	404
Using the JCL? Command to Create a Batch Job	407
Using the Distributed JCL to Create a Batch Job	410
Creating Your Own Batch Job	411
How to Submit a Batch Job	412
Options for Batch Jobs	413
Changing the Value of Batch Options	415
Building Profile Variable Input for Batch Processing Using PROFOUTE	417
Additional Considerations for Batch Processing	418

Planning for Running Batch Jobs

Overview

This unit describes the prerequisite activities you should perform before running a batch job. The activities apply whether you use the JCL? command to generate your batch job, modify the examples !DB/EXPLAIN provides, or write your own batch job.

Options for creating a batch job

To create a batch job, you can either:

- Issue the JCL? command from any !DB/EXPLAIN panel. !DB/EXPLAIN presents you with the panels necessary to build the JCL to run your job.
 - Copy and modify the example JCL !DB/EXPLAIN provides
 - Write your own batch job.
-

Prerequisites to running a batch job

Before running a batch job:

- Review and modify the JCL distributed with !DB/EXPLAIN to reflect the values specific to your installation.
 - Review the options specified on your Housekeeping panels to confirm that the values set during installation meet your needs.
 - Identify the objects on which you want to perform an action.
 - Understand which commands, selects, and filters are available to act on that object.
 - If you are using the example JCL, locate and modify the appropriate examples and include them in your job stream.
 - If you are writing your own batch job, create the statements required to execute the job.
-

Defining your region size to run batch

Candle Corporation recommends that you define a TSO and batch region size of at least 4 megabytes for any !DB/Tools product.

Determining Which Batch Job to Use

Overview

This unit describes the types of batch jobs available with !DB/EXPLAIN.

Types of batch jobs

There are two types of batch jobs:

1. Jobs that run against “nonextracted” entities such as SQL statements in a dataset, BIND statements, or DBRMs. Such a job could be used to:
 - EXPLAIN the entity
 - Obtain exceptions or recommendations for the entity
 - View statements

The user most often performing these kinds of tasks is the application programmer who wants to review the behavior of SQL code in an application before the application is bound.

2. A job that runs against extracted entities; for example, a Catalog extract and PLAN_TABLE extract. This type of job can be used to perform most of the tasks you can perform online. It typically supports a user performing the task of database management or administration.

You use JCL members shipped in the !DB/EXPLAIN *hilev*.CNTL dataset to execute these jobs (see “Example JCL and Control Members” on page 404).

Example JCL and Control Members

Overview

This unit describes the JCL members and control members used to execute !DB/EXPLAIN in batch mode and provides the criteria to help you decide which batch job to execute.

Background about the example JCL and control members

Example JCL and control members are shipped in the !DB/EXPLAIN *hilev.CNTL* dataset.

Example JCL

The chart below describes the batch jobs shipped with !DB/EXPLAIN:

JCL Member	Description
<i>DB2idUTIL</i>	Provides the basic JCL to do most !DB/EXPLAIN functions in batch. Note: When !DB/EXPLAIN generates JCL for this job, it prefixes the job name with the ID of the current DB2 subsystem. When !DB/EXPLAIN identifies the job in a panel example, it does so using the identifier KTEBUTIL.
KTEBDBR2	The “batch equivalent” of KTEXPLD, it EXPLAINS all SQL statements in DBRMs.
KTEBDBRM	The “batch equivalent” of KTEXPLB, it EXPLAINS the SQL statements contained in the DBRM members to be included in the plan or package if the plan or package were bound.
KTEBSQLS	The “batch equivalent” of the online edit macro KTEXPL, it EXPLAINS the SQL statements in a file.

Note: For information about the interactive !DB/EXPLAIN edit macros KTEXPL, KTEXPLD, and KTEXPLB referenced in the chart, see “Accessing and Exiting !DB/EXPLAIN” on page 69.

Control members shipped with !DB/EXPLAIN

In addition to the JCL listed above, the following members are shipped in the !DB/EXPLAIN *hilev*.CNTL library. The table below describes the contents of these members:

Control Member	Description
KTESYSIN	Contains examples of tasks and respective commands that run against extracted entities. You can select examples that perform the tasks you want to do and include these statements as part of the SYSIN input to your <i>DB2idUTIL</i> job.
KTESYSIS	Contains examples of tasks and respective commands that run against nonextracted entities. You can select examples that perform the tasks you want to do and include the appropriate statements as part of the SYSIN input to your KTEBSQLS, KTEBDBRM, or KTEBDBR2 job.
KTECNTL	Contains statements used by your batch job to override profile variable specifications. !DB/EXPLAIN runs your batch job using the profile variables from your profile data set unless you specifically override them. To temporarily override profile variables, modify the statements contained in KTECNTL. (You can override any profile variable except those specified on the Housekeeping panel SQL Parameters for KTEXPL.)
KTEPFVAR	Documents the profile variables used by the !DB/EXPLAIN batch JCL. KTEPFVAR defines each variable and identifies the source of the variable's value. For example, the variable EXP_BUILD_STAT is defined as “whether to display BUILD status during initialization” and its source is the Extract Processing Defaults Housekeeping panel.

Selecting the correct batch job

The table below contains a list of some common tasks you can perform using !DB/EXPLAIN. While the list is not exhaustive, it should help you select the appropriate batch JCL to perform the tasks your installation requires. Detailed information on performing these and other !DB/EXPLAIN batch tasks can be found in the sections referenced in column 3.

Task I Want to Perform	Batch Job to Use	Find Details In:
EXPLAIN DBRMs	KTEBDBR2	“Batch Processing for Application Development” on page 419
EXPLAIN BIND commands (all DBRMs that would be included in the plans, packages, or both that would be bound by the BIND commands)	KTEBDBRM	“Batch Processing for Application Development” on page 419
EXPLAIN the SQL statements in a dataset or imbedded in a program	KTEBSQLS	“Batch Processing for Application Development” on page 419
EXPLAIN an extracted plan, package, DBRM, or statement	<i>DB2idUTIL</i>	“Batch Processing for Database Administration” on page 439
Generate a BIND for an extracted plan or package	<i>DB2idUTIL</i>	“Batch Processing for Database Administration” on page 439
View cost data for extracted plans or packages	<i>DB2idUTIL</i>	“Batch Processing for Database Administration” on page 439
View or compare EXPLAIN histories	<i>DB2idUTIL</i>	“Batch Processing for Database Administration” on page 439

Using the JCL? Command to Create a Batch Job

Overview

This unit provides information about using !DB/EXPLAIN's JCL? command and its associated panel-driven interface to create a batch job.

Background about the JCL? command

Using the JCL? command is the easiest and fastest way to create a batch job to perform !DB/EXPLAIN activities. You can issue the JCL? command from any !DB/EXPLAIN panel.

Procedure for using the JCL? command

Follow these steps to use the JCL? command.

Step	Action
1	Type JCL? on the command line of any !DB/EXPLAIN panel.
2	Press Enter. Result: !DB/EXPLAIN displays the JCL Generation panel.
3	<p>On the JCL Generation panel, type the number of the job you want to create in the Option field.</p> <pre data-bbox="573 1157 1248 1486"> ----- DB/EXPLAIN DB2=D31A ----- Cmd ==> J C L G E N E R A T I O N Option ==> 1 DB/EXPLAIN Batch Utility (KTEBUTIL) 2 Explain SQL statements in a data set (KTEBSQLS) 3 Explain BIND commands in a data set (KTEBDBRM) 4 Explain SQL statements in a DBRM (KTEBDBR2) 5 Bind DB/EXPLAIN 6 DB/EXPLAIN Extract </pre>

Procedure for using the JCL command (continued)

Step	Action
4	Press Enter Result: !DB/EXPLAIN displays one of the following panels based on the option you chose: <ul style="list-style-type: none"> ● Generate !DB/EXPLAIN Batch Utility ● Generate JCL for Processing SQL Statements ● Generate JCL for Processing BIND Command ● Generate JCL for Processing DBRMS ● Generate !DB/EXPLAIN BIND Commands ● Extract JCL Generation
5	On the secondary panel: <ul style="list-style-type: none"> ● Specify values for required and optional fields such as objects to be acted on, dataset names, whether to use member lists, whether to alter the KTECNTL or SYSIN control statements, and so forth. ● Provide job statements to run the job or alter existing job statements.
6	Press Enter. Result: !DB/EXPLAIN generates the JCL required to run your job.
7	!DB/EXPLAIN displays the Output Options panel, on which you can specify the name of the dataset and member where the JCL is to be stored. The default is member JCL in the user PDS.
8	To accept the default, Press Enter. Result: !DB/EXPLAIN stores the JCL in the user PDS and displays the Output JCL Options panel on which you can choose to submit the job, print or edit the JCL, defer the job, or cancel the job.

Background about using the KTEMODEL command

If you select any of options 1 through 4 on the JCL Generation panel, you can use the KTEMODEL command on the secondary panel to create model SYSIN input.

Using the KTEMODEL command

Follow this process to use the KTEMODEL command to create model SYSIN input.

Step	Action
1	On the Generate JCL panel: <ul style="list-style-type: none"> Specify the values you want. Type Y in the Edit Commands (SYSIN) field.
2	Press Enter. Result: !DB/EXPLAIN displays a pop-up listing the commands available.
3	Type the option for the command you want to use.
4	Press Enter. Result: !DB/EXPLAIN displays the Edit SYSIN panel.
5	Type KTEMODEL on the command line.
6	Type the appropriate ISPF line command too insert the JCL for for the model scenario. <ul style="list-style-type: none"> To insert the JCL after the line, type A. To insert the JCL before the line, type B.
7	Press Enter. Result: !DB/EXPLAIN displays a panel listing the model scenarios available.
8	Type the number for the model scenario you want on the command line.
9	Press Enter. Result: !DB/EXPLAIN redisplay the Edit SYSIN panel, with the JCL for the model scenario in the position you specified.

Using the Distributed JCL to Create a Batch Job

Overview

This unit suggests ways you can use the JCL distributed with !DB/EXPLAIN to make you more productive when creating and executing a batch job.

Background about the JCL

An alternative way to create a batch job to perform !DB/EXPLAIN activities is to copy and modify the JCL and example tasks distributed with !DB/EXPLAIN. Each JCL member contains information about the modifications you must make to customize it for your installation. The control members KTESYSIN and KTESYSIS contain commented examples of common !DB/EXPLAIN tasks.

Printing the example JCL

You may want to print out and review the example JCL to prepare for:

- Customizing it for your installation
- Copying and/or modifying the example control statements
- Using it as a model for creating your own batch jobs

To do so, you can use the ISPF print facility or a printing process specific to your installation.

Creating Your Own Batch Job

Overview

This unit provides some background information about creating your own batch job.

Background about creating your own batch job

If you want to perform !DB/EXPLAIN functions that are not provided in the distributed examples, you can create your own !DB/EXPLAIN batch job.

Because the number and type of batch control statements to complete a task differ depending upon whether the task is performed against extracted or nonextracted entities, details for writing your own batch job are found in “Batch Processing for Database Administration” on page 439 and “Batch Processing for Application Development” on page 419 respectively.

Definition of extracted and nonextracted entities

An *extracted entity* is one that contains data that has been extracted from the DB2 Catalog and, optionally, PLAN_TABLE. The data is associated with a specific extract ID. A plan is an example of an extracted entity.

A *nonextracted entity* is one for which the data to be explained resides in the entity itself rather than in an !DB/EXPLAIN extract. Examples of a nonextracted entity include a program containing SQL statements or a DBRM.

Before you begin

Determine what function you want your batch job to perform. If you are familiar with running online !DB/EXPLAIN, determine how you might perform the function online. Then translate the job flow to batch control statements.

How to Submit a Batch Job

Overview

This unit provides information on submitting batch jobs from ISPF.

Submitting a batch job from ISPF

Be sure you meet the requirements for running batch jobs. Then, follow these steps to submit a job from ISPF.

Step	Action
1	Access ISPF and open the member in the user PDS that contains the JCL for your batch job.
2	On the ISPF command line, type SUBMIT or SUB .
3	Press Enter.

Options for Batch Jobs

Overview

This unit describes the way in which !DB/EXPLAIN uses profile variables as options for batch jobs.

Determining a Housekeeping option's associated profile variable

Each !DB/EXPLAIN Housekeeping option has an associated profile variable. You can determine the profile variable associated with a given field on a Housekeeping panel by reviewing the online help for the panel. Typically, the information that cross-references a specific Housekeeping field to an associated profile variable is contained in the last help panel in a help sequence.

A reverse index listing !DB/EXPLAIN's profile variables in alphabetical order and cross-referencing them to a specific field on a specific Housekeeping panel is located in the *hilev.CNTL* dataset member KTEPFVAR.

Value of batch options at !DB/EXPLAIN start-up

When you initiate a batch job, the values of the batch job options are the default values of !DB/EXPLAIN's profile variables.

Determining batch option values at start-up

To determine the default value for a given profile variable:

- Review the online help available with !DB/EXPLAIN's Housekeeping panels to determine the value
 - Look up the profile variable name in member KTEPFVAR to determine the Housekeeping panel on which the value is set
-

Displaying the contents of KTEPFVAR

You can display the contents of member KTEPFVAR by:

- Typing PROFVAR on the command line of any !DB/EXPLAIN panel
 - Entering option 0 on the !DB/EXPLAIN Primary Menu
-

How !DB/EXPLAIN sets values for a given batch execution

When a batch job is initiated, !DB/EXPLAIN follows the process described in the following table to determine the options applicable to a given batch execution.

Step	Action
1	At start-up, !DB/EXPLAIN assumes the defaults for all profile variables.
2	!DB/EXPLAIN reads the profile datasets to determine whether you have chosen to override any of the defaults (by means of permanent changes made via Housekeeping panels). If you have entered any overrides, !DB/EXPLAIN accepts them.
3	During the batch run, !DB/EXPLAIN reads the KTECNTL input to determine whether you want to override any options listed in the profile datasets. If so, !DB/EXPLAIN accepts them. Only those profile variables that begin with EXP_ can be overridden via KTECNTL input.

Supporting installation standards by controlling overrides

In order to control the use of overrides in your installation, you may want to do one of the following:

- Create a dataset containing the KTECNTL options you want to use in your installation, name it (for example, **MY.KTECNTL**), and define it as:

```
//KTECNTL DD DISP=SHR,DSN='MY.KTECNTL'
```

- Create separate datasets, one containing the profile option overrides for online sessions and one containing the profile option overrides for batch executions. Then, instead of using KTECNTL input, simply point to the batch profile DSN.

Obsolete profile variables

From time to time, profile variables are obsoleted by a new !DB/EXPLAIN version. If you enter an obsolete variable, you receive warning message KTE096. !DB/EXPLAIN ignores the input command and continues with processing. If you receive warning message KTE096, remove the override specification for this variable from your KTECNTL input.

Changing the Value of Batch Options

Overview

This unit describes the ways in which you can change the value of the options that affect your batch job.

Changing initial values

You can change !DB/EXPLAIN defaults in one of these ways:

- Updating relevant fields on the Housekeeping panels. If you have appropriate authority, you can permanently save the updates by specifying **PERMANENT=Y** on the Housekeeping panels. Values you have chosen to permanently retain are stored in the profile data set.
 - Overriding values by including override commands as part of your KTECNTL input
-

Guidelines for creating your KTECNTL input

When entering override commands using KTECNTL:

- Ensure that the variable statement you are including in the input begins with **EXP_**. Only these profile variables can be overridden.
 - Ensure there are no imbedded blanks in the profile variable command. The = must not have blanks either before or after it.
 - Specify numeric values in normal arabic numbers. Do not use hexadecimal format.
 - Place an asterisk (*) in column 1 to indicate that the input on that line is a comment. The line is ignored.
 - If you enter a blank line, !DB/EXPLAIN ignores it.
-

Example of an override command

The example below specifies an override using KTECNTL input commands:

```
//KTECNTL DD *  
EXP_MSG_SUP=Y
```

Generating a large volume of override commands

If you want to override a large number of profile variables, you can use the PROFOUTE command to generate your KTECNTL input. For information, see the next unit, “Building Profile Variable Input for Batch Processing Using PROFOUTE” on page 417.

Building Profile Variable Input for Batch Processing Using PROFOUTE

Overview

This unit tells you how to use the command PROFOUTE to create an input job stream to override a large number of profile variables.

Result of issuing the command PROFOUTE

You can use the command PROFOUTE to build a KTECNTL input job stream that can be used for batch processing. When you issue the command PROFOUTE from the command line during an !DB/EXPLAIN session, !DB/EXPLAIN writes all of the EXP_* profile variables to your user PDS in a format useable in a KTECNTL job stream. These profiles variables are written to a member named *userid*||P.

Format of PROFOUTE output

PROFOUTE writes the !DB/EXPLAIN profile variables to your user PDS in !DB/EXPLAIN external format. If you want to have the profile variables written to your user PDS in internal format, which can be used to diagnose problems, use the command PROFOUT instead of PROFOUTE. For information about PROFOUT, see “Using the PROFOUT Command” on page 463 in the chapter entitled “Resolving Problems.”

Saving your PROFOUTE output

PROFOUTE and PROFOUT both write their output to the user PDS member *userid*||P. If you use both of these commands during an !DB/EXPLAIN session, the output of the second command you issue will overwrite the output already contained in your user PDS member. To save the output of the first command, copy it to another appropriately named member.

Procedure for using the PROFOUTE command

Step	Action
1	Type the command PROFOUTE on the command line of a !DB/EXPLAIN panel.
2	Press Enter. Result: !DB/EXPLAIN writes the current setting of the user profile variables that begin with EXP_ to the user PDS member named <i>userid</i> P.

Additional Considerations for Batch Processing

Overview

This unit describes some factors you should consider when using the !DB/EXPLAIN batch facility. It refers you to sections of this manual where expanded information can be found.

Using the CEXPL command in batch

When you want to compare a historic EXPLAIN with its current counterpart, you issue the CEXPL command. When used in batch, CEXPL has some restrictions. For information and an example of the use of CEXPL, see “Compare History” in the *!DB/EXPLAIN Reference*.

Running batch and online !DB/EXPLAIN sessions concurrently

You can run !DB/EXPLAIN online and batch sessions concurrently. However, there are performance implications to doing so. Various Housekeeping variables can be set to provide the most advantageous environment for concurrent access. These variables are discussed in the Appendix entitled “Running !DB/EXPLAIN Concurrently in Batch and Online” on page 477.

Chapter 22.

Batch Processing for Application Development

Introduction

This chapter describes the batch facilities !DB/EXPLAIN provides to assist you in developing high-performance applications. Before using this chapter, you should be familiar with some basic concepts that apply to batch processing with !DB/EXPLAIN. These can be found in the chapter “Using !DB/EXPLAIN in Batch” on page 401.

This chapter describes in detail the input commands you can use against nonextracted entities, entities for which the data to be EXPLAINED resides in the entity itself rather than in a !DB/EXPLAIN extract. Detailed information on using selects and macros in batch are contained in “Batch Processing for Database Administration” on page 439.

Chapter contents

JCL for Batch Application Development	420
Modifying the JCL—Common Parameters	423
Modifying the JCL—Parameters Specific to KTEBSQLS	425
Modifying the JCL—Parameters Specific to KTEBDBR2	427
Modifying the JCL—Parameters Specific to KTEBDBRM	428
Input to Your Batch Job	429
Putting It Together—Flow of your Batch Job	432
Bypassing the Statements Display in Batch	437

JCL for Batch Application Development

Overview

This unit describes the batch jobs shipped with !DB/EXPLAIN to support the task of application development and maps the tasks you want to perform against the batch jobs used to perform those tasks.

Background about !DB/EXPLAIN batch processing and application development

An application developer can use a batch job shipped with !DB/EXPLAIN and stored in the CNTL library to:

- Review statements related to a nonextracted entity

A nonextracted entity is one for which the data to be explained resides in the entity itself and not in a !DB/EXPLAIN extract. The entity might be, for example, a program containing SQL statements, a BIND statement containing DBRMs, or a DBRM.
 - EXPLAIN the entity
 - Obtain exception reports and recommendations related to the entity
-

Using the JCL? command to create batch JCL

The easiest and fastest way to create the JCL to run the batch jobs that support application development is to use the JCL? command. You can issue the JCL? command from the command line of most !DB/EXPLAIN panels. For information about using the JCL? command, see “Using the JCL? Command to Create a Batch Job” on page 407 in the chapter “Using !DB/EXPLAIN in Batch.”

Using the !DB/EXPLAIN Online Menu, KTEON, to create batch JCL

The application development batch jobs permit you to EXPLAIN SQL statements and BIND commands in physical sequential or partitioned data sets, and DBRMs in partitioned data sets. You can use !DB/EXPLAIN's Online Menu to create the JCL to execute these jobs. On the Online Menu, you can specify the name of the data set to be EXPLAINed. If you are using partitioned data sets, you can process multiple data sets by naming them on a secondary panel as part of the JCL generation process. You can process multiple members by naming them on the INCLUDE or EXCLUDE parameters in the JCL. You can also create a list of data set members to process using a pattern mask or list of pattern masks. To understand how to provide values for the INCLUDE and EXCLUDE parameters, see the three units later in this chapter that contain specification examples. To review information about creating member name pattern masks, see "Access Using the EXEC KTEON" on page 90.

What JCL should I use?

Use this table to determine the name of the batch job to perform your !DB/EXPLAIN application development tasks.

Task I Want to Perform	Batch Job to Use
EXPLAIN the SQL statements in a data set or imbedded in a program	KTEBSQLS
EXPLAIN DBRMs	KTEBDBR2
EXPLAIN BIND commands (all DBRMs that would be included in the plans, packages, or both that would be bound by the subject BIND commands).	KTEBDBRM

Modifying the JCL—Common Parameters

Overview

This unit describes the modification or customization you must perform on the parameters common to the batch jobs KTEBSQLS, KTEBDBR2, and KTEDBRM to tailor them to your installation.

Common parameters

This table describes the modifications required to parameters that are common to all three batch jobs.

Parameter	Value You Supply
DB2LOAD=	Name of your DSNLOAD data set
LOAD= and LOAD()	Name of the data set containing the !DB/EXPLAIN load library
CLST=	Name of the library containing the !DB/EXPLAIN CLISTs
MSGS=	Name of the !DB/EXPLAIN messages library
SOUT=	SYSOUT class for messages
ROUT=	SYSOUT class for reports
UPROF=	User profile data set name
SPROF=	System profile data set name
DB2()	Extract or DB2 subsystem For nonextracted entities, the extract ID you point to is the one that you configured with the desired options for the DB2 subsystem you want to run against.
DSN()	Name of the data set(s) containing the DBRMs, BIND commands, or SQL statements. Data sets must be either a sequential file or partitioned data set(s). The format for the data set name is: DSN(this.data.set)

Common parameters (continued)

Parameter	Value You Supply
INCLUDE()	If the DSN parameter identifies a partitioned data set, this parameter specifies one or more member names or member pattern masks to be processed. The format for the INCLUDE parameter is INCLUDE(member mask1 mask2)
EXCLUDE()	If the DSN parameter identifies a partitioned data set, this parameter specifies one or more member names or member pattern masks to be excluded from processing. The format for the EXCLUDE parameter is EXCLUDE(member mask1 mask2)
VIO()	Symbolic name for virtual I/O (specific to the current MVS system)
BLKSIZE()	Maximum block size to use when allocating VIO data sets. To use the system-determined block size, enter 0 .
ESTAE()	Debug flag

Processing order for INCLUDE and EXCLUDE

As !DB/EXPLAIN executes the JCL, INCLUDE is processed first. Then that list is processed to perform the EXCLUDE.

Modifying the JCL—Parameters Specific to KTEBSQLS

Overview

This unit describes the modification or customization you must perform on the parameters specific to batch job KTEBSQLS to tailor it to your installation.

KTEBSQLS-specific parameters

This table describes the modifications required to the parameter specific to KTEBSQLS.

Parameter	Value You Supply
FORMAT()	<p>Format of the data set containing the SQL statements. Possible format values are:</p> <p>A Assembler</p> <p>B C</p> <p>C COBOL</p> <p>F FORTRAN</p> <p>P PLI</p> <p>Q QMF</p> <p>S SPUFI</p> <p>T, U, V, W, or X Undefined; may be user-defined</p> <p>Note: Your installation may have tailored the values A-S to represent user-defined formats other than those shown above.</p>

Example JCL to EXPLAIN SQL statements

This JCL EXPLAINS SQL statements in the data set MY.USER.SOURCE, in the members whose member names start with SQ. However, that list is further subsetted to exclude any members whose name is SQL or whose name starts with the letters SQL plus any other single character.

```
//EXBATCH.SYSTSIN DD *
%KTEXPLAB LOAD(DB.EXPLAIN.LOAD) +
          DB2(DB2) +
          DSN(MY.USER.SOURCE) +
          INCLUDE(SQ*) +
          EXCLUDE(SQL SQL%) +
          FORMAT(S) +
          VIO(VIO) +
          BLKSIZE(3200) +
          ESTAE(Y)
```

This example modifies the job shown above to process multiple partitioned data sets.

```
//EXBATCH.SYSTSIN DD *
%KTEXPLAB LOAD(DB.EXPLAIN.LOAD) +
          DB2(DB2) +
          DSN( +
            MY.USER.SOURCE +
            MY.GROUP.SOURCE +
            MY.PROD.SOURCE +
          ) +
          INCLUDE(SQ*) +
          EXCLUDE(SQL SQL%) +
          FORMAT(S) +
          VIO(VIO) +
          BLKSIZE(3200) +
          ESTAE(Y)
```

Modifying the JCL—Parameters Specific to KTEBDBR2

Overview

This unit describes the modification or customization you must perform on the parameters specific to batch job KTEBDBR2 to tailor it to your installation.

KTEBDBR2-specific parameters

This table describes the modifications required to parameters specific to KTEBDBR2.

Parameter	Value You Supply
PLAN()	Name of the Plan (used only to generate reports)
OWNER()	Plan owner; used as a qualifier for unqualified names.

Example JCL to EXPLAIN DBRMs

This JCL EXPLAINS DBRMs in the data set MY.USER.DBRM, in the members whose member names start with DBRM, KTE, or MY followed by any number of additional characters. However, the list excludes members that begin with the letters DBRM01 or DBRM03. Also excluded are member names that begin with DBRM, end with the character 2, and have one intervening character; and member names that begin with KTE, end with the characters 23, and have any number of intervening characters.

```
//EXBATCH.SYSTSIN DD *
  %KTEXPLDB LOAD(DB.EXPLAIN.LOAD) +
    DB2(DB2) +
    DSN(MY.USER.DBRM) +
    INCLUDE(DBRM* KTE* MY*) +
    EXCLUDE(DBRM01* DBRM03* DBRM%2 KTE*23) +
    OWNER(MYLOGON) +
    VIO(VIO) +
    BLKSIZE(3200) +
    ESTAE(Y)
```

Modifying the JCL—Parameters Specific to KTEBDBRM

Overview

This unit describes the modification or customization you must perform on the parameters specific to batch job KTEBDBRM to tailor it to your installation.

KTEBDBRM-specific parameters

There are no parameters specific to KTEBDBRM. However, you can include a //DBRMLIB DD statement in your JCL. The data set or data sets specified by this statement are searched for the DBRMs specified in the BIND commands. If the DBRMs are not found in the data sets specified in the allocated DBRMLIB DD statement, the data sets named in the LIBRARY() parameter for the BIND commands or the DBRMLIB DD of the BIND command's JCL are searched next.

Example JCL to EXPLAIN BIND commands

This JCL EXPLAINS SQL statements in BIND commands in library DB.EXPLAIN.LOAD. Members included are any whose names start with the letters BIND followed by any number of additional characters. However, the list excludes members that begin with the letters BINDJ followed by any number of additional characters.

```

//EXBATCH.SYSTSIN DD *
%KTEXPLBB LOAD(DB.EXPLAIN.LOAD) +
          DB2(DB2) +
          DSN(MY.USER.JCL) +
          INCLUDE(BIND*) +
          EXCLUDE(BINDJ*) +
          VIO(VIO) +
          BLKSIZE(3200) +
          ESTAE(Y)
    
```

Input to Your Batch Job

Overview

This unit describes the inputs you can include in your batch job to accomplish !DB/EXPLAIN tasks.

SYSIN input

Input for your batch jobs is included as commands in the SYSIN input deck. Subsequent sections describe the command types that you can include in SYSIN. When working with nonextracted entities, the most frequently used commands are:

- Commands—!DB/EXPLAIN commands you can issue from a panel and utility commands such as PRNT and END
 - Filters
-

Overriding Profile Variables

A number of factors controlling the processing of your batch job are specified by means of parameters on !DB/EXPLAIN's Housekeeping panels and stored in a profile data set. You can override most of these profile variables by including the appropriate statements in the KTECNTL JCL. Candle Corp. suggests you avoid manipulating profile variables until you are very familiar with !DB/EXPLAIN, with the results manipulation of the variables can cause, and the interaction of various variables with each other.

Adding a comment to SYSIN input commands

Review the chart for the method to add a comment to a batch input command.

Task You Want to Perform	Action
Add a comment to an input command	Type an asterisk (*) as the first character of each line of the comment. (You cannot continue a comment line using a plus sign.)

Commands

Two command types are available for use in batch:

Batch utility commands

Commands used to control batch utility operations; for example PRNT (print) or END. These commands help you manage your !DB/EXPLAIN jobs.

!DB/EXPLAIN commands

Commands specific to !DB/EXPLAIN that act on !DB/EXPLAIN objects. A list of !DB/EXPLAIN-specific commands is found in “Commands Available from Object List Panels” in the *!DB/EXPLAIN Reference*.

Most commands you can issue online against a selected object can also be issued in batch. (Commands that require user interaction such as GLOBAL, DO, FILT?, FORMAT? and SORT? are not available in batch.)

Entering commands

To enter a command, simply include the command as a line of SYSIN input, for example:

OUT (From the Statements display, write output to a dataset)

or

PRNT (Print output as a report)

Further examples of the use of commands against nonextracted entities are found in “Putting It Together—Flow of your Batch Job” on page 432.

Filtering commands

You can apply a filter to an object set to limit the number of objects affected by subsequent processing. A full discussion of filtering is found in “Using Filters” on page 147. To use a filter in batch, include a filter statement in your SYSIN input command grouping. Possible filter choices are:

Filter specification	Result
FILTA	<i>And</i> filtering
FILTR	<i>Or</i> filtering
FILTO	Turn off filtering; retain settings
NOFILT	Reset all filters

Immediately following FILTA or FILTR, include commands identifying the objects to be filtered; for example, you might use the following input command groupings for the Statements display:

```

FILTR (set OR filter)
SEXPL=Y (returns: EXPLAINable statements or
SCURS=ACURSOR cursor name is ACURSOR)

FILTA (set AND filter)
SEXPL=Y (returns: EXPLAINable statements having
SSTNO>500 statement nbr. greater than 500)

```

Additional examples of the use of a filter against nonextracted entities are found in “Putting It Together—Flow of your Batch Job” on page 432.

Putting It Together—Flow of Your Batch Job

Overview

This unit describes the flow of a batch job and provides examples you can use to understand !DB/EXPLAIN batch processing and as models for batch jobs you want to create.

Background about creating a batch job to run against nonextracted entities

To review, you can build a batch job that operates on a nonextracted entity to:

- Review SQL statements residing in a nonextracted entity such as a data set or application program
- EXPLAIN the entity
- Obtain exception reports and recommendations

In order to construct your batch job most efficiently, you should think about how you might do a given task online and then convert the actions to batch commands. This unit helps you understand how to do so.

Reviewing statements

When you run the batch job, the first objects you deal with are statements (unless your profile indicates that statement processing is to be bypassed. If you have specified that statement processing be bypassed, see the unit “Bypassing the Statements Display in Batch” on page 437 later in this chapter to understand the effect on batch processing flow.)

To process statements, you can issue the same commands that are available to you on the online Statements display.

Statement examples

The table below describes some tasks you can perform on statements and includes example SYSIN input. Remember, you can use only those commands available from the Statements display.

Purpose	Example
Print all the statements.	SHOW (Expand all statements) PRNT (Output to REPORT DD) END (Terminate processing)
Print all of the EXPLAINable statements.	FILTR (Indicates a filter) SEXPL=Y (Only look for EXPLAINable statements) SHOW (Expand the filtered statement list) PRNT (Output to REPORT DD) END (Terminate processing)
Print a report for the statements showing all exceptions.	XCPT (Identify exceptions) PRNT (Print exception report) END (Terminate processing)
For EXPLAINable statements only, print the statements display and output the statements to a file.	FILTR (Indicates a filter) SEXPL=Y (Only look for EXPLAINable statements) PRNT (Output to REPORT DD) OUT (Write to file) END (Terminate processing)

Note: The OUT command writes its output to the user PDS whose member name you specify on the Output Options Housekeeping panel. !DB/EXPLAIN stores the member name in the profile variable **EXP_SQL_MBR**.

Generating EXPLAINS

You can also generate EXPLAINS of your SQL statements, including those associated with BIND commands and DBRMs.

EXPLAIN examples

Here are some examples of generating and reporting on EXPLAINS. Remember, you can use only those commands available from the EXPLAIN display.

Purpose	Example
EXPLAIN all statements. Do not gather statistics. Print the EXPLAIN report.	EXPLA (EXPLAIN/do not gather stats.) PRNT (Output to REPORT DD) END (Terminate processing)
EXPLAIN all statements. Gather statistics. Output the EXPLAIN report.	GEXPLA (EXPLAIN/gather stats.) EXODS (Output to dataset) END (Terminate processing) <i>Note:</i> The EXODS command writes its output to the user PDS whose member name you specify on the Output Options Housekeeping panel. !DB/EXPLAIN stores the member name in the profile variable EXP_EXODS_MBR .

Formatting your EXPLAIN Display

You can control the format of your EXPLAIN display by reviewing and, if required, altering the parameters entered on Housekeeping's EXPLAIN Display Tuning Parameters panel. Information about this topic is found in "Controlling the Format of the EXPLAIN Display" on page 303.

Combination examples

Remember that, when planning your batch jobs, you need to think of how you would perform the task interactively using !DB/EXPLAIN's online panels. This example walks you through a multi-step process using commands available from the Statements and the EXPLAIN displays.

When you issue a command that takes you to a new display, you can issue the commands available to you on that display. Using the END command returns you to a previous display. When you have returned to the previous display, you can only enter commands valid there.

Example 1

In combination example 1, for the following SYSIN input stream:

```
FILTR
SEXPL=Y
PRNT
EXPLA
PRNT
END
OUT
END
```

the processing flow is:

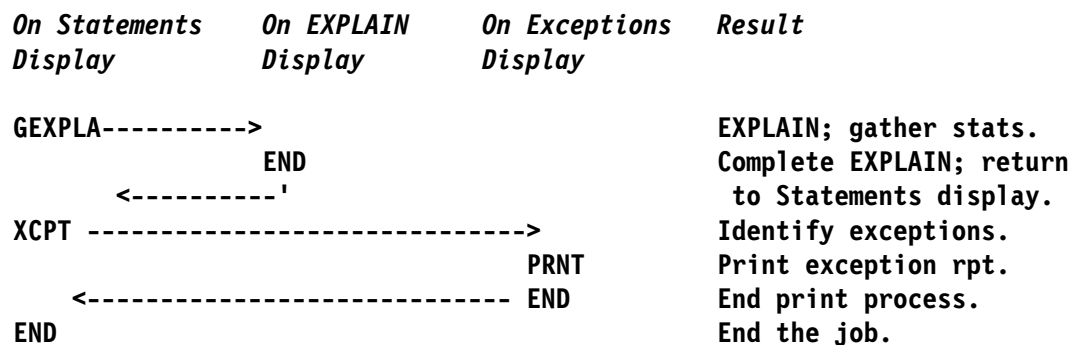
<i>On Statements Display</i>	<i>On EXPLAIN Display</i>	<i>Result</i>
FILTR		Filter for
SEXPL=Y		EXPLAINable statements.
PRNT		Print the statements.
EXPLA ----->		EXPLAIN the statements.
	PRNT	Print the EXPLAIN report.
	END	End processing on EXPLAIN
	<-----'	display; return to Stmts display.
OUT		Output statements to file.
END		End the job.

Example 2

In combination example 2, for the following SYSIN input stream:

```
GEXPLA
END
XCPT
PRNT
END
```

the processing flow is:



Bypassing the Statements Display in Batch

Overview

This unit describes the effect on a batch job of specifying that the Statements display should be bypassed.

Causing the statements display to be bypassed

You determine whether the Statements display is to be bypassed by the value you specify on Housekeeping's SQL Defaults panel in the field Bypass Statements Display?. Three specifications are possible:

- | | |
|----------|--|
| N | Do not bypass the Statements display. |
| Y | Bypass the Statements display. Perform an EXPLAIN and do not gather statistics (equivalent to specifying EXPLA on the Statements display). |
| G | Bypass the Statements display. Perform an EXPLAIN and gather statistics (equivalent to specifying GEXPLA on the Statements display). |

The value you specify on the SQL Defaults panel is placed into the profile variable **EXP_BYPASS_STMT**.

Result of the specification

Specifying **Y** or **G** has the effect during batch processing of causing Statements processing to be omitted. Therefore, you *cannot* use Statements panel commands as part of your batch SYSIN input.

For example, if **EXP_BYPASS_STMT=G**, then including the following command grouping in your SYSIN input:

```
PRNT  
END
```

would print the EXPLAIN, gather statistics, and end the job.

Bypassing the Statements Display in Batch.

Chapter 23.

Batch Processing for Database Administration

Introduction

This chapter describes the batch facilities !DB/EXPLAIN provides to assist you in identifying and modifying problem SQL code, comparing historical and current EXPLAINS, and managing your DB2 objects. The batch commands discussed in this chapter operate on extracted data from the DB2 Catalog or PLAN_TABLE. Before using this chapter, you should be familiar with some basic concepts that apply to batch processing with !DB/EXPLAIN. These can be found in the chapter “Using !DB/EXPLAIN in Batch” on page 401.

Chapter contents

JCL for Batch Database Administration and Management	440
Input to your Batch Job	442
A Comprehensive Example	452

JCL for Batch Database Administration and Management

Overview

This unit describes the JCL for !DB/EXPLAIN's batch utility and tells how to modify and use it.

Background about the batch utility JCL

A database administrator can use the batch utility shipped with !DB/EXPLAIN and stored in the CNTL library to perform virtually all of the tasks that can be performed using !DB/EXPLAIN's online panels. When !DB/EXPLAIN generates JCL for this job, it prefixes the job name with the ID of the current DB2 subsystem. When !DB/EXPLAIN identifies the job in a panel example, it does so using the identifier KTEBUTIL. *DB2idUTIL* operates against data that has been extracted from the DB2 Catalog and, optionally, PLAN_TABLE and associated with a specific extract ID. Available tasks include:

- EXPLAINing various DB2 objects
 - Comparing EXPLAINs
 - Comparing EXPLAIN history
 - Obtaining exception reports and recommendations
 - Reviewing and comparing costs
 - BINDing and REBINDing objects
-

Creating the JCL to run the batch utility

The easiest and fastest way to create the JCL to run the batch utility, *DB2idUTIL*, is to execute the JCL command from the command line of any !DB/EXPLAIN panel. For information about using the JCL command, see “Using the JCL? Command to Create a Batch Job” on page 407 in the chapter “Using !DB/EXPLAIN in Batch.”

Modifying the batch utility JCL

Review this table to understand the modifications required to these parameters to tailor *DB2idUTIL* to your installation.

Parameter	Value You Supply
DB2LOAD=	DSNLOAD
LOAD=	Name of the data set containing the !DB/EXPLAIN load library
CLST=	Name of the library containing the !DB/EXPLAIN CLISTs
MSG=	Name of the !DB/EXPLAIN messages library
SOUT=	SYSOUT class for messages
ROUT=	SYSOUT class for reports
UPROF=	User profile data set name
SPROF=	System profile data set name
Extract or DB2 subsystem	Extract ID the batch job will reference
ESTAE	Debug flag

Input to Your Batch Job

Overview

This unit describes the inputs you can include in your batch job to accomplish !DB/EXPLAIN tasks.

Before you begin

Determine what function you want your batch job to perform. If you are familiar with running online !DB/EXPLAIN, you may want to work out how you might perform the function online. Then translate the job flow to batch commands.

Create a SYSIN deck containing the command groupings to perform the activities you define: Each group of commands:

- *Must* start with a command specifying the type of object to be operated on by all subsequent filters and commands
- Specifies the commands to be used against the object
- Specifies the selects to be used against the object
- Identifies the filters to be applied
- Identifies the sorts to be used
- Identifies the formats to be used
- May contain macros provided by !DB/EXPLAIN or written by you and saved in your User PDS
- Ends with the END command

SYSIN input

Input for your batch jobs is included as commands in the SYSIN input deck. Subsequent sections describe the command types that you can include in SYSIN.

Commands (actions to be performed) must be contained within a primary command grouping. The primary commands are those that identify objects.

Overriding Profile Variables

A number of factors controlling the processing of your batch job are specified by means of parameters on !DB/EXPLAIN's Housekeeping panels and stored in a profile data set. You can override some of these profile variables by including the appropriate statements in the KTECNTL JCL. Candle Corp. suggests you avoid manipulating profile variables until you are very familiar with !DB/EXPLAIN, with the results manipulating the variables can cause, and with the effect variable interactions with each other can cause.

Tasks frequently performed when coding batch input commands

Review the chart for the method to perform common tasks when creating batch input commands.

Task You Want to Perform	Action
Add a comment to a command.	Type an asterisk (*) as the first character of each line of the comment. (You cannot continue a comment line using a plus sign.)
Apply the subsequent command to each selected object.	Type a forward slash (/) between the functions. Example: S P/REBIND
Continue a line of a command to the next line.	Type a plus sign (+) as the last character of the line that is being continued.

What are primary commands?

Primary commands are commands that identify the !DB/EXPLAIN object to be acted upon by subsequent secondary commands in a command grouping. The format of a command grouping is typically:

```
Primary Command
.
.  n secondary commands
.  (action commands)
.
END
```

Objects represented by primary commands

The primary commands and the objects they represent are in the following table:

Command	Object Selected
PL	Plan
DB	DBRM
KA	Package
CL	Collection
TA	Table
LI	Library
CO	Cost
XC	Exception
HI	Extract History

Secondary batch commands

Secondary commands can be of two types:

Batch utility commands

Commands used to control batch utility operations; for example PRNT (print) or END. These commands help you manage your !DB/EXPLAIN jobs. They can also move you back and forth between “displays”.

!DB/EXPLAIN commands

Commands specific to !DB/EXPLAIN that act on !DB/EXPLAIN objects. !DB/EXPLAIN-specific secondary commands are commands that can be issued only on the display with which they are associated (that is, against the specific objects they act on). These commands can also move you back and forth between displays. A list of !DB/EXPLAIN-specific commands and the objects they act on are found in “Commands Available from Object List Panels” in the *!DB/EXPLAIN Reference*.

There is no limit to the number of commands that can follow the object command.

Commands that are not available in batch

Most commands you can issue online against a selected object can also be issued in batch. Commands that require user interaction such as GLOBAL, DO, FILT?, FORMAT? and SORT? are not available in batch.

Command examples

The chart below provides some examples of primary and secondary commands used in batch jobs.

Purpose	Example
EXPLAIN and gather statistics on all of the packages which have not been EXPLAINed since they were bound.	KA (Primary cmd.: Object = Packages) GEXPLO (Command: EXPLAIN/gather stats) PRNT (Command: Output to REPORT dd) END (Terminate the example) Note: The PRNT command output is sent wherever you have defined in your REPORT dd JCL statement—this may be a JES class or a physical data set.
Print the Information panel for plans. This example prints all of the INFO panels for plans beginning with ABC.	PL (Primary cmd.: Object = Plans) FILTA (Indicates a filter) PLAN=ABC* (Only plans starting with ABC) INFO (Command: Display info for plans meeting criteria) PRNT (Command: Output to REPORT dd) END (Terminate the example)

Moving between displays using commands

The example below illustrates how secondary commands cause you to move from !DB/EXPLAIN display to display to create a report.

<i>Primary</i>	<i>Plans</i>	<i>DBRMs</i>	<i>Cost</i>
<i>Display</i>	<i>Display</i>	<i>Display</i>	<i>Display</i>
PL ----->	PRNT		
	DBRMS----->	PRNT	
		COST----->	PRNT
			END
		PRNT <-----'	
		END	
	PRNT <-----'		
	END		
<-----'			

The output printed report would contain in this order:

- The plans list
- The DBRMs list
- The costs list
- The DBRMs list
- The plans list

Selects

You can use most !DB/EXPLAIN selects in batch. To simulate a select in batch, type **S** *x* where *x* is the select character you would have typed when using !DB/EXPLAIN online. A list of the selects available to you cross-referenced to the object lists to which they apply are found in “Selects Available from Object List Panels” in the *!DB/EXPLAIN Reference*.

Select example

The chart below provides an example of using a select in batch.

Purpose	Example
Display all tables that the DBRM <i>dbrmname</i> accesses.	DB (Object = DBRMS) FILTA (Indicates a FILTER) DBRM=<i>dbrmname</i> (Only DBRMs named <i>dbrmname</i>) S T/PRNT/END (Select and report Tables for each DBRM) END (Terminate the example)

Note: The example above contains two **END** commands. The first **END**, issued from the TABLES display, terminates the print job and returns you to the DBRMS display. The second **END** terminates the example command set.

Selects can also be nested in batch. To nest selects in batch, use number (#) signs as the second-level function delimiters to indicate the inner nesting of the second select and PRNT relative to the first select.

The chart below provides an example of using a nested select in batch.

Purpose	Example
Display all tables that the DBRM <i>dbrmname</i> accesses and all SQL statements that access these tables.	DB (Object = DBRMS) FILTA (Indicates a FILTER) DBRM=<i>dbrmname</i> (Only DBRMs named <i>dbrmname</i>) S T/PRNT/S S#PRNT#END/END END (Terminate the example)

Filters

You can apply a filter to an object set to limit the number of objects affected by subsequent processing. There is no limit to the number of filters which can follow the starting object command.

Filtering example

The table that follows illustrates the use of filters within a statement group.

Purpose	Example
Filter plans, display DBRMs in the plans, filter the resultant DBRMs, and print them.	PL (Object = Plans) FILTA (Indicates a filter) PLAN=<i>planame</i> (Only Plans named <i>planame</i>) DBRMS (Select DBRMs) FILTA (Indicates another filter) DBRM=<i>dbrmname</i> (Only DBRMs named <i>DBRMname</i>) PRNT (Output to REPORT dd) END (Terminate the example)

Macros

You can use macros within command groupings in a batch job. “Using the Macro Facility” on page 183 describes example macros shipped with !DB/EXPLAIN and tells you how to create your own macros. Example commands illustrating the use of a macro can be found in the CNTL library member KTESYSIN.

Macro example

The table that follows illustrates the use of a macro within a command group.

Purpose	Example	
Display all tables that the DBRM <i>dbrmname</i> accesses. (Assumes that the commands in the example shown under “Selects” above are contained in macro \$distab .)	DB \$DISTAB END	(Object = DBRMs) (Macro containing the following statements: FILTA DBRM=<i>dbrmname</i> S T/PRNT/END (Terminate the example)

Masks

To achieve better performance, you may want to limit the number of objects you manipulate. In this case, it is preferable whenever possible to use the masks defined in the KTECNTL statements rather than to specify a filter or filters as part of your batch statement group.

Masking is different from filtering in that data that is “masked out” is not present in storage. Data that is “filtered out” is present in storage, but is not displayed to you.

During a batch session, any masking that is in effect is that which you have specified on Housekeeping's Selection Masking panel. You can override these values by modifying the applicable profile variables in KTECNTL.

Formatting the EXPLAIN display

You can control the format of your EXPLAIN display by reviewing and, if required, altering the parameters entered on Housekeeping's EXPLAIN Display Tuning Parameters panel. Information about this topic is found in “Controlling the Format of the EXPLAIN Display” on page 303.

A Comprehensive Example

Overview

This example illustrates the use of masks, filters, and commands in a batch run.

Example input

This example rebinds all plans that start with the letter **B** and contain statements to perform tablespace scans. The example control statements are input to the batch utility *DB2idUTIL*.

```
//EXBATCH.KTECNTL DD *
EXP_PLAN_OPER=LIKE
EXP_PLANMASK=B%
//EXBATCH.SYSIN DD *
CO
FILTA
C#TSS>0
S P/REBIND/END
END
```

!DB/EXPLAIN's response

Here is !DB/EXPLAIN's response to the example control statements.

Statement	!DB/EXPLAIN Response
//EXBATCH.KTECNTL DD *	Batch control card
EXP_PLAN_OPER=LIKE EXP_PLANMASK=B%	Masks for only those plans starting with the letter B .
//EXBATCH.SYSIN DD *	Batch control card
CO	Identifies Costs as the object type being operated on. Plans to be considered are only those beginning with B .
FILTA	Turns on <i>and</i> filtering.
C#TSS>0	Filters for plans that perform a tablespace scan.
S P/REBIND/PRNT/END	Selects the plans that passed the filtering criterion, rebinds them, outputs a list of the selected plans to REPORT DD, and ends the activity.
END	Terminates the statement group.

A Comprehensive Example

Problem Resolution

Chapter 24. Resolving Problems

Introduction

This chapter describes the resources available to assist you in resolving problems. It includes instructions for responding to error and problem messages. It provides directions for contacting Candle Customer Support should the need arise, and for creating a diagnostic report for them if you are requested to do so. Additional diagnostic techniques are found in the Appendix “Additional Problem Resolution Techniques” on page 467.

Chapter contents

Determining the Level of Your !DB/EXPLAIN System . . .	458
!DB/EXPLAIN Online Messages	459
Assistance from Candle Customer Support	461
Using the PROFOUT Command	463

Determining the Level of Your !DB/EXPLAIN System

Overview

This unit provides information to assist you in determining the level of your !DB/EXPLAIN system.

Procedure to determine the maintenance level of your system

The maintenance level of your !DB/EXPLAIN system is displayed on the !DB/EXPLAIN Primary Menu. If you want to determine the maintenance level from another !DB/EXPLAIN panel, follow the steps in this chart.

Step	Action
1	To obtain the maintenance level of the !DB/EXPLAIN system you are using, type MAINT on the command line of a !DB/EXPLAIN panel.
2	Press Enter. Result: !DB/EXPLAIN displays the maintenance level of the system you are using.

Procedure to determine the maintenance level of your !DB/Tools

The maintenance level of all of the installed !DB/Tools can be displayed by following the steps in this chart.

Step	Action
1	To obtain the maintenance level of every installed component of !DB/Tools, type MAINT? on the command line of a !DB/EXPLAIN panel.
2	Press Enter. Result: !DB/EXPLAIN displays the maintenance level of the !DB/Tools you have installed.

!DB/EXPLAIN Online Messages

Overview

This unit describes the facilities !DB/EXPLAIN provides to assist you in resolving problems.

Online help available to resolve problems

!DB/EXPLAIN displays messages, their explanations, and appropriate user actions, if any, online during product execution. If an error or problem occurs, !DB/EXPLAIN displays a short error message in the upper right corner of the menu or panel. A number may precede the short text of the error message.

Extended Help

Many error messages have extended Help. Extended help can comprise long message text and, for some messages, additional error message panel displays. To view extended Help, follow the steps in this chart.

Step	Action
1	On the panel that is displaying the short message in the upper right corner of the screen, type HELP on the command line, or press the appropriate function key.
2	Press Enter. Result: !DB/EXPLAIN displays the long message if it is available.
3	On the panel that is displaying the long message type HELP on the command line, or press the appropriate function key.
4	If an extended Help panel exists, !DB/EXPLAIN displays it.

Example of short and long messages

This chart shows two levels of detail available from !DB/EXPLAIN for online Help.

Initial Display When the Error Occurs	Display When You Use the HELP Command (Long Message)
No explain data.	KTE021K No previous explain data available to compare bind results to.

Using the MSGHELP Command to directly access extended Help

You can display extended help directly if you know a message number (for example, KTE021K). You may want to do this when, for example, you receive a short numbered message in a batch run and want to see the online information about the error. Follow the directions in this chart to use MSGHELP.

Step	Action
1	On the command line of any !DB/EXPLAIN panel, type MSGHELP .
2	Press Enter. Result: !DB/EXPLAIN displays the Message Help panel.
3	On the Message Help panel, type the message identifier of the message you want to see extended Help for in the Message Identifier field. Result: !DB/EXPLAIN displays the extended Help for the message whose identifier you entered.

Assistance from Candle Customer Support

Overview

This unit provides information to help you contact and use the expertise of Candle Customer Support. It also provides you with information about diagnostic facilities available to assist you in gathering information required by Candle Customer Support.

Actions to obtain support

If an error message instructs you to call Candle Customer Support or if you encounter a problem requiring assistance, call Candle Customer Support toll free at 1-800-328-1811. (More complete instructions on obtaining support from Candle Customer Support are found in “Introduction” on page 483.) Every effort will be made to resolve your problem immediately. This unit provides you with instructions for obtaining some of the information that Candle Customer Support may ask you to provide.

Providing Information to Aid in Problem Diagnosis

For most error conditions, Candle Customer Support will want to see a copy of your activity log and a printout of the contents of the !DB/EXPLAIN data spaces and control blocks.

Printing the activity log

Before you call Candle Customer Support regarding a product ABEND, please print a copy of the activity log.

Printing the activity log allows you to have on hand the information Candle Customer Support needs and will speed up the resolution of the problem. Usually, Candle Customer Support will ask you to fax the hard copy of the activity log. Follow these steps to print the activity log.

Step	Action
1	On the !DB/EXPLAIN Administration Menu, type 1 in the Options field. Result: The system displays the Log Activity List panel.
2	Move the cursor to the line with the ABEND whose log you want to print.
5	Type P in the Select field.
6	Press Enter. Result: The system displays a pop-up that allows you to specify print options.
7	Type the print options you want.
8	Press Enter. Result: The system processes the print request and prints the activity log at the destination you specified.

Printing the contents of !DB/EXPLAIN data spaces and control blocks

When you send the activity log printout to Candle Customer Support, you should also send a printout of the contents of !DB/EXPLAIN data spaces and buffers at the time the ABEND occurred. This information is automatically written to SYSOUT(x) if you specified **Use Dataspace=Y** on Housekeeping's Miscellaneous Defaults panel.

If the type of error you experience is not an ABEND, Candle Customer Support may request that you issue one of the SNAP commands. In addition, some of the extended Help requests that you issue one of the SNAP commands before calling Candle Customer Support. See "Using the SNAP Commands" on page 468 for information about using the SNAP commands.

Using the PROFOUT Command

Overview

This unit describes the PROFOUT command and how you can use it during a !DB/EXPLAIN session.

Purpose of using PROFOUT

You can use the PROFOUT command to diagnose problems, should any occur during a !DB/EXPLAIN session. When you issue the PROFOUT command during a !DB/EXPLAIN session, !DB/EXPLAIN writes the current setting of every user profile variable known to !DB/EXPLAIN to the !DB/EXPLAIN User partitioned dataset. This dataset has the member name *userid*||P.

Format of PROFOUT output

PROFOUT writes the !DB/EXPLAIN profile variables to your user PDS in !DB/EXPLAIN internal format. If you want to have the profile variables written to your user PDS in external format, which can be used as part of your KTECNTL input for a batch job, use the command PROFOUTE instead of PROFOUT. For information about PROFOUTE, see “Building Profile Variable Input for Batch Processing Using PROFOUTE” on page 417 in the introductory chapter about batch processing.

Saving your PROFOUT output

PROFOUT and PROFOUTE both write their output to the user PDS member *userid*||P. If you use both of these commands during an !DB/EXPLAIN session, the output of the second command you issue will overwrite the output already contained in your user PDS member. To save the output of the first command, copy it to another member.

Procedure for using the PROFOUT command

To use the PROFOUT command, follow the steps in this chart.

Step	Action
1	Type the command PROFOUT on the command line of a !DB/EXPLAIN panel.
2	Press Enter. Result: !DB/EXPLAIN writes the current setting of every user profile variable known to it to the user PDS member named <i>userid</i> P.

Appendixes

Appendix A.

Additional Problem Resolution Techniques

This appendix provides some additional methods for debugging !DB/EXPLAIN and providing input to Candle to assist in problem resolution.

When to use the techniques in this appendix

Complex or unique SQL may yield unexpected results and require additional diagnostic information to understand the way in which DB2 interprets it. The techniques in this appendix *are for diagnostic purposes and should be used when so directed by Candle Customer Support.*

Where to find more information

Use the table below to find information about additional debugging aids.

Task you want to perform	Source of information
Using the SNAP commands to write !DB/EXPLAIN information to the KTESNAP data set.	p.468
Turning traces on and off from the !DB/EXPLAIN command line	p.470
Interactively tracing statement execution	p.471
Using the automated trace facility with the DEBUG commands	p.472
Diagnosing parser interpretation with the PDEBUG commands	p.474

Using the SNAP Commands

Overview

This unit describes the use of the SNAP commands to produce diagnostic information to assist in resolving any problems that can occur when executing !DB/EXPLAIN.

Use of the SNAP commands

The SNAP commands are diagnostic commands that cause !DB/EXPLAIN data areas to be written to the KTESNAP DD data set.

Command	Description
SNAPEX	Writes the current !DB/EXPLAIN control blocks to KTESNAP.
SNAPDS	Writes the contents of !DB/EXPLAIN data spaces and control blocks to KTESNAP.
SNAPALL	Writes all problem program information to the KTESNAP data set (SNAPX PDATA=ALL). The !DB/EXPLAIN control blocks and data spaces are also written to KTESNAP.
SNAPALLS	Writes all system control program information and all problem program information to KTESNAP (SNAPX SDATA=ALL PDATA=ALL). The !DB/EXPLAIN control blocks and data spaces are also written to KTESNAP.

Procedure for using SNAP commands

Under certain circumstances, Candle may request you to submit the KTESNAP output as part of your problem report. Follow this procedure to obtain the output.

Step	Action
1	Type the command SNAPx on the command line of a !DB/EXPLAIN panel, where x may be: <ul style="list-style-type: none"> ● EX ● DS ● ALL ● ALLS Review the command definitions in the section above to understand what information is written by each of the SNAP commands.
2	Press Enter. Result: !DB/EXPLAIN writes the required information to KTESNAP.

Routing of your SNAP output

If you are running !DB/EXPLAIN online, your KTESNAP output is routed to SYSOUT(x).

If you are running in batch, your KTESNAP output is routed to the class specified in KTESNAP DD.

Using the TRACE Commands

Overview

This unit identifies commands provided to trace various !DB/EXPLAIN functions.

Background about the trace commands

Tracing can be resource-intensive. For diagnostic purposes, and when so directed by Candle, use the trace commands identified in this unit to trace various !DB/EXPLAIN and system functions. Candle Customer Support may request that you provide this input when placing a request for service. Enter the appropriate trace command by typing it on the command line of any object list panel.

Data sets for trace output

The output of a trace command is sent to a data set identified as follows:

- When initiated during an online session, the TRACEDD name is dynamically generated. It takes the form TE $nnnnn$, where $nnnnn$ is a system-generated value.
 - When initiated from a batch session, trace output is written to KTBTRAC.
-

Available trace commands

Task You Want to Perform	Command to Use
Trace getmains, freemains, data space functions, and list functions.	TRACEALL
Trace data space functions.	TRACED
Trace getmains and freemains.	TRACEG
Trace list functions.	TRACEL
Turn tracing off.	TRACEOFF

Interactively Tracing Statement Execution

Overview

This unit describes the procedure to interactively trace statement execution.

Procedure to trace statement execution

When so directed by Candle, use the steps in this chart to gather additional diagnostics. The method used is to reset the ESTAE flag when accessing !DB/EXPLAIN using the edit macro KTEXPL.

Step	Action
1	In an edit session, open the member you want to EXPLAIN.
2	On the command line, enter: KTEXPL <i>extract_id</i> <i>format</i> D <i>extract_id</i> Is the extract ID you are using <i>format</i> Is a letter representing the data format of the member you are EXPLAINing; for example, C for COBOL or S for SPUI D Affects ESTAE processing by turning on an interactive trace
3	Press Enter. Result: The REXX EXECs are interactively traced line by line. After each line is displayed, press Enter to display the next line.
4	To speed processing, you can enter TRACE OFF and press Enter. (You may have to repeat this action several times.) !DB/EXPLAIN then processes the balance of EXEC and displays the SQL statement as sent to DB2.
5	Press Enter. Result: When a non-0 SQL code is returned, the statement is displayed as it was sent to DB2. Send copies of the original SQL test and the redisplayed text to Candle Customer Support.

Using the Automated Trace Facility

Overview

This unit describes the automated trace facility and provides instructions for its use.

Background about the automated trace

A trace facility is provided that can capture information useful to Candle personnel in resolving problems. Snapshots of storage and register values are written to a file at strategic times during the execution of !DB/EXPLAIN

Note: !DB/EXPLAIN ATF is resource intensive. Use it only when working with Candle personnel to resolve a problem.

Prerequisite to using the automated trace facility in batch

In order to use the automated trace facility in batch, add the following DD statement to your batch JCL:

```
//KTCTRACE DD SYSOUT=*
```

Using the automated trace facility in batch

All DEBUG commands used in batch processing must be used within primary commands. This example opens KTCTRACE and turns on the trace facility.

```
PL      (Primary Command)  
DEBUGON  
PRNT  
END     (Primary Command)
```

Tracing begins and continues until !DB/EXPLAIN encounters a DEBUGOFF command. All trace information is written to the trace data set.

Using the automated trace facility online

The table that follows tells you how to turn the automated trace facility on and off and how to use it.

Step	Action
1	<p>Issue the command DEBUGON from the command line. Optionally include a limit counter on the number of program traces. For example:</p> <p style="text-align: center;">CMD==> DEBUGON 100 (100 traces performed)</p> <p>or</p> <p style="text-align: center;">CMD==> DEBUGON (default limit is 500)</p> <p>Result: The system turns on the trace facility.</p>
2	<p>Tracing continues until:</p> <ul style="list-style-type: none">● The specified trace limit is reached, or● You enter the command DEBUGOFF on the command line
3	<p>When the trace facility is turned off either through expiration of the trace limit or entry of DEBUGOFF, !DB/EXPLAIN writes trace information to the output class you have specified for the field SYSOUT Hold Class on Housekeeping's !DB/Tools Global Configuration Information panel.</p>

Diagnosing Parser Interpretation

Overview

This unit describes the PDEBUG commands that are used to diagnose parser interpretation.

Background about the PDEBUG commands

There are two commands that permit you to turn parser interpretation on and off.

- PDEBUGON sets parser debugging on
 - PDEBUGOFF sets parser debugging off
-

Using the PDEBUG commands online

When so directed by Candle, use the steps in this chart to gather additional diagnostics. Candle Customer Support may request that you provide this input when placing a request for service.

Step	Action
1	To turn parser debugging on, enter the command PDEBUGON on the command line of any !DB/EXPLAIN panel.
2	Press Enter. Result: Debugging begins. Output is routed to SYSOUT class X as the default.
3	To turn debugging off, enter PDEBUGOFF on the command line of an !DB/EXPLAIN panel.
4	Press Enter. Result: Debugging ends.

Requirement for using the PDEBUG commands in batch

In order to use the PDEBUG commands in batch, you must first add the following DD statement to your batch JCL:

```
//PDEBUG DD SYSOUT=&SOUT,RECFM=FB,LRECL=100,BLKSIZE=1000
```

Examples of using the PDEBUG commands in batch

Example 1: If you want to debug an entire series of jobs, set parser debugging on using a KTECNTL override statement as shown below. Then include all of the jobs you want to debug in your SYSIN input.

```
//KTECNTL DD *  
EXP_PARSE_DEBUG=Y  
/*  
//SYSIN DD *  
PL  
EXPLA  
END
```

Example 2: If you want to debug just a single job step, you can create input such as this:

```
PL  
PDEBUGON  
EXPLA  
PDEBUGOFF  
END
```


Appendix B. Running !DB/EXPLAIN Concurrently in Batch and Online

This appendix provides some guidelines you can use if you want to run concurrent batch and online !DB/EXPLAIN sessions.

Restriction on concurrent extract

You cannot run concurrent extracts to the same set of extract files in !DB/EXPLAIN.

Background about running concurrent batch and online sessions

When !DB/EXPLAIN performs an EXPLAIN, it opens the statements extract dataset to retrieve the statements. Each statement is passed individually to DB2 to be explained. DB2 places the results in the DB2 PLAN_TABLE. !DB/EXPLAIN queries the DB2 PLAN_TABLE for the results and updates its PLAN_TABLE extract dataset.

Variables affecting concurrent access to extract datasets

Variables on Housekeeping's Tuning Parameters panel affect the process of ENQUEUE/ALLOCATE/OPEN/CLOSE/DEALLOCATE/DEQUEUE that !DB/EXPLAIN performs against its extract (catalog, statement, PLAN_TABLE) datasets. This table describes the relevant Housekeeping tuning parameters that affect concurrent access.

Variables affecting concurrent access to extract datasets (continue d)

Housekeeping Variable	Value Set to Y	Value Set to N
Enqueue on extract datasets	<p>!DB/EXPLAIN issues an EXCLUSIVE enqueue on the catalog and statement extract datasets for a catalog extract, or on the PLAN_TABLE extract dataset for a plan table extract.</p> <p>In the online environment, during the build process, !DB/EXPLAIN issues a SHARE enqueue for all three datasets to read the history/restart record. This ensures that the datasets are in sync and that no extract is running. If the datasets are synchronized and no extract is running, !DB/EXPLAIN immediately releases the ENQUEUE on the statements extract dataset. !DB/EXPLAIN releases the ENQUEUE on the catalog extract dataset after completing the build. It releases the ENQUEUE on the PLAN_TABLE extract dataset after reading cost data from it.</p> <p>During an update (XUPDT) to the PLAN_TABLE extract dataset, !DB/EXPLAIN issues an EXCLUSIVE enqueue. It releases the enqueue at the completion of the EXPLAIN function or at a commit point.</p>	!DB/EXPLAIN does not enqueue on any dataset
Deallocate statements extract dataset after each use?	Each time !DB/EXPLAIN accesses the statements extract dataset, it closes and deallocates it.	!DB/EXPLAIN allocates and opens the statements extract dataset at first use and closes and deallocates it at the end of the session.
Deallocate PLAN_TABLE extract dataset after each use?	Each time !DB/EXPLAIN accesses the PLAN_TABLE extract dataset, it closes and deallocates it.	!DB/EXPLAIN allocates and opens the PLAN_TABLE extract dataset at first use and closes and deallocates it at the end of the session.

Variables affecting concurrent access to extract datasets (continued)

Housekeeping Variable	Value Set to Y	Value Set to N
Keep PLAN_TABLE extract dataset open for update?	<p>!DB/EXPLAIN keeps the PLAN_TABLE extract dataset allocated and open.</p> <p>If the variable Deallocate PLAN_TABLE extract dataset after each use? is set to Y, a Y value for this variable is ignored.</p> <p>If you assign a value of F to this variable, !DB/EXPLAIN keeps the PLAN_TABLE dataset open for the duration of the command or select being performed (the duration of the EXPLAIN(s)).</p>	Close the PLAN_TABLE extract dataset after each commit point.

Controlling the frequency of commit points

To provide for greater concurrency, you can control the frequency with which !DB/EXPLAIN issues a commit. Use these two Housekeeping variables, found on the Tuning Parameters panel, to control the frequency with which !DB/EXPLAIN issues a COMMIT.

- Number of DBRMs to process before freeing intermediate storage
- Number of statements to process before freeing intermediate storage

The smaller number of DBRMs or statements processed before !DB/EXPLAIN frees resources, the greater the possible concurrency. Too-frequent commit points can impact concurrency, while too few commit points can impact recoverability. Thus, you will need to experiment to find the optimum value for your site.

Recommendations for Concurrent Access

Overview

This unit provides Candle Corp.'s recommendations for the values to be applied to the Tuning Parameter variables if you want to perform concurrent batch and online access of !DB/EXPLAIN.

Background about the recommendations

The recommendations in this unit assume that the dominant use of !DB/EXPLAIN is to perform any of the various EXPLAIN functions, for example, commands GEXPLA, EXPLA, CEXPL, or LEHIST, or selects E or L, online in real time.

The recommendations further assume that the batch functions are performed during off hours when online activity is minimal or absent. !DB/EXPLAIN's batch functions can be particularly useful when EXPLAINing long lists of objects or updating your production subsystem by adding new plans or binding/rebinding existing plans. Using batch minimizes the effect of these processes on your production environment.

Setting defaults

You can specify default values for the four Tuning Parameter variables under discussion by specifying them on the Housekeeping Tuning Parameters panel and making them permanent (saving them to the profile dataset).

Online environment recommendations

To achieve the best combination of performance, concurrency, and data integrity during an online session, Candle Corp. suggests that you set the Housekeeping variables as shown in this table.

Housekeeping Variable	Recommended Value	Rationale
Enqueue on extract datasets	Y	Specifying Y provides for good data integrity. Concurrency is impacted in that an extract cannot be run when KTE is in use. In a high-use environment, occasional enqueue conflicts on the PLAN_TABLE extract dataset can occur.
Deallocate statements extract dataset after each use?	N	The overhead of allocating and opening the dataset is encountered only at startup rather than for each access to the statements extract dataset. Specifying N impacts concurrency since this value makes it more difficult to allocate the dataset with DISP=OLD. !DB/EXPLAIN does not attempt to allocate the dataset as OLD except to perform a delete/define. Concurrency among online users is not impacted because the dataset is opened for read from within KTE.
Deallocate PLAN_TABLE extract dataset after each use?	N	This dataset is opened for output. The overhead of allocation and open is encountered for the first statement EXPLAINed. Concurrency is impacted only for the length of time it takes to perform a function.
Keep PLAN_TABLE extract dataset open for update?	F	Using F causes the PLAN_TABLE extract dataset to remain open for the length of the EXPLAIN function. To EXPLAIN a single object, the time during which the dataset is open is dependent on the size of the object and the performance of DB2 and MVS. If a list of objects is being EXPLAINed, the dataset can be open for an extended period. During this period, no other client can update the PLAN_TABLE extract dataset using an EXPLAIN. If you require greater concurrency than provided by the use of F, consider using N as the value for this variable. You can also assess the frequency of commit points as discussed earlier in this appendix.

Batch environment recommendations

Batch recommendations are different from online recommendations. Using the recommended values provides better throughput, but limits multiple concurrent users. Refer to the table for batch recommendations.

Housekeeping Variable	Recommended Value
Enqueue on extract datasets	Y
Deallocate statements extract dataset after each use?	N
Deallocate PLAN_TABLE extract dataset after each use?	N
Keep PLAN_TABLE extract dataset open for update?	Y

Overriding default values for batch processing

In order to ensure that the batch utility *DB2idUTIL* runs with the appropriate values for the variables, you need to provide the following control cards in your KTECNTL input.

Housekeeping Variable	Control Card to Override Default Profile
Enqueue on extract datasets	EXP_ENQUEUE=Y
Deallocate statements extract dataset after each use?	EXP_STMT_DEALLO=N
Deallocate PLAN_TABLE extract dataset after each use?	EXP_PTBL_DEALLO=N
Keep PLAN_TABLE extract dataset open for update?	EXP_XUPDT_OPEN=Y

Appendix C. Candle Customer Support

Introduction

Candle Corporation offers a comprehensive maintenance and support plan to ensure you realize the greatest value possible from your Candle software investments. We have more than 200 technicians worldwide, committed to providing you with prompt resolutions to your support requests.

Customer Support hours of operation are from 5:30 A.M. to 5:00 P.M., Pacific Time. In the event of an after-hours or weekend emergency, Candle's computerized call management system ensures that a technician will return your call within one hour. For customers located outside of North America, after-hours and weekend support is provided by Candle Customer Support locations in the United States.

Electronic Support

Candle provides information and support services using

- Candle's home page at www.candle.com. You can use the Candle Web site to
 - open problem records
 - access maintenance information
 - order products or maintenance
 - access IBM compatibility information
 - download fix packs for distributed products
 - read news and alerts
 - scan a list of scheduled Candle education classes
- Candle Electronic Customer Support (CECS), an electronic customer support facility. You can access this facility through the IBM Global Network. You can use CECS to
 - open problem records
 - search our database for solutions to known problems
 - look for answers to commonly asked questions
 - read news and alerts
 - scan a list of scheduled Candle education classes

Both CECS and the Candle Web site are available 24 hours a day, 7 days per week.

Telephone Support

Our support network consists of product specialists who work with you to solve your problem.

Candle uses an online problem management system to log and track all support requests. Your request is immediately routed to the appropriate technical resource.

When you call to report a problem, please have the following information:

- your Candle personal ID (PID) number
- the release level of the Candle product
- the release level of IBM or other vendor software
- identifying information and dates of recently applied maintenance to your Candle product or IBM product
- a detailed description of the problem (including the error message) and the events preceding the problem
- a description of any unusual events that occurred before the problem

Customer Support Phone Numbers

	Telephone	Fax
North America	(800) 328-1811	
	(310) 535-3636	(310) 727-4204
Europe		
Belgium/Luxembourg	+32 (0) 3 270 95 60	+32 (0) 3 270 95 41
France	+33 (0) 1 53 61 60 60	+33 (0) 1 53 61 06 16
Germany/Switzerland/ Austria	+49 (0) 89 54 554 333	+49 (0) 89 54 554 170
Italy - Freephone	800 780992	
Netherlands	+31 (0) 30 600 35 50	+31 (0) 30 600 35 10
Scandinavia	+46 (0)8 444 5940	+46 (0)8 623 1855
U.K.	+44 (0)161 437 5224	+44 (0)161 437 5225
(Southern Europe, Middle East and South Africa Agents call U.K.)		
Asia Pacific - English Hub		+61 2 9954 1818
Australia	+61 2 8912 9898	
Hong Kong	800 908 457	
India	+61 2 8912 9898	
Indonesia	0018 03061 2061	
Malaysia	1800 803 459	
New Zealand	0800 449 596	
Philippines	1800 1612 0096	
Singapore	800 616 2075	
Thailand	0018 00612 1045	
Asia Pacific - Japanese Hub	+81 3 3595 7150	+81 3 3595 7110
Asia Pacific - Korean Hub	+82 2 552 8744	+82 2 552 8746
Asia Pacific - Mandarin Hub	+88 62 2739 3223	+88 62 2378 5993
Asia Pacific e-mail address: ap_support@candle.com		

When your local support office is unavailable, you can contact Candle's North America support center. If USADirect® service is available in your country, use the 800 telephone number. If USADirect service is not available, ask your international operator for assistance in calling Candle's local (310) number.

Incident Documentation

You may be asked to send incident documentation to the Candle Customer Support Center. On the outside of all packages you send, please write the incident number given to you by the Customer Support representative.

Send tapes containing the incident information to the following address, unless directed otherwise by your Customer Support representative:

Candle Customer Support
Candle Support Center, *Incident number*
201 North Douglas Street
E1 Segundo, CA 90245

Send all other relevant documentation, such as diskettes or paper documentation, to the address provided by your Customer Support representative.

Ensuring Your Satisfaction with Customer Support

Candle Customer Support is committed to achieving high customer satisfaction ratings in all areas. These include

- connecting you to a support representative promptly
- providing you with the appropriate fixes
- answering support questions
- filling your shipping orders
- supplying documentation

If you have a concern that has not been resolved to your satisfaction, you can open a complaint ticket. All tickets are logged and tracked to ensure responsiveness and closure. Using the ticket information, a manager will contact you promptly to resolve your problem.

Index

Special Characters

- _ 156
- []
 - documentation conventions 17
- { }
 - documentation conventions 17
- &\$CNT variable
 - on Output Options Defaults panel 319, 322
- %
 - documentation conventions 16

A

- access
 - concurrent batch and online 477
 - from German QMF bridge 87
 - from KTEON 90
 - from KTEQMF 87
 - from KTEQMFGE 87
 - from KTEXPLA 90
 - from KTEXPLBA 90
 - from KTEXPLD 90
- access, concurrent
 - overriding default values
 - input to KTECNTL job 482
 - variables affecting 477
 - recommendations 480
 - recommendations for batch environment 482
 - recommendations for online environment 481
 - setting defaults 480
- accessing Administration Menu 110
- accessing !DB/EXPLAIN 69
 - and displaying statements 101
 - background about 70
 - by invoking !DB/EXPLAIN CLIST 73
 - from an edit macro 77
 - from !DB/Tools Product Selection Menu 71
 - from KTEON 90
 - from KTEXPL 78
 - from KTEXPLA 92
 - from KTEXPLB 86

- accessing !DB/EXPLAIN (*continued*)
 - from KTEXPLBA 95
 - from KTEXPLD 98
 - methods for 70
 - on first execution 76
- accessing !DB/EXPLAIN indirectly
 - CLIST for OMEGAMON II for DB2 104
- accessing JCL Generation Menu 112
- accessing Primary Menu 108
- activities, basic 67
- activity log
 - accessing from Administration Menu 111
 - accessing using ACTLOG command 111
 - used by customer support 462
- additional sessions
 - filtering in 159
- ADDSYS option
 - used to propagate !DB/EXPLAIN 343
- ADF 78
- Administration Menu
 - accessing 110
 - accessing from Primary Menu 109
 - elements of 110
- Adobe portable document format 14
- ALIAS extract function option 364
- appendixes 465
- Application Developer Facility 78
- application development
 - KTEXPL used in 78
 - performing remote EXPLAIN 224
 - performing remote EXPLAINS 224
 - remote EXPLAIN 224
- assembler 78
- ATF 472
- Automated Trace Facility 472

B

- backup 395
- basic functions 67
- batch 401
 - concurrent access with online session 477
 - recommendations 480
 - variables affecting 477

Index

- batch (*continued*)
 - parser debugging in 475
 - restriction on CEXPL command 418
 - Batch Automated Trace Facility 472
 - batch commands
 - for nonextracted jobs 430
 - SUB 412
 - SUBMIT 412
 - batch example macros 188
 - batch input commands
 - adding a comment 430, 443
 - applying a subsequent command to each object 443
 - continuing a line to the next line 430, 443
 - for nonextracted jobs 430
 - batch JCL 404
 - batch job
 - changing batch options 415
 - control members shipped with !DB/EXPLAIN 405
 - creating your own 411
 - determining which to use 403
 - example JCL 404
 - extracted entities
 - definition 411
 - for extracted entities 403
 - for nonextracted entities 403
 - initiating 413
 - jobs shipped with !DB/EXPLAIN 404
 - main discussion 401
 - nonextracted entities
 - definition 411
 - options at !DB/EXPLAIN start-up 413
 - options for 413
 - prerequisite discussion 401
 - prerequisites for 402
 - profile variables 413
 - region size for 402
 - submitting 412
 - batch jobs
 - using JCL? command to create 407
 - batch masking 348
 - batch processing
 - concurrent with online session 418
 - batch trace facility 472
 - batch utility
 - creating JCL for 440
 - batch, using 399
 - beginning !DB/EXPLAIN 69
 - bind, remote 227
 - BookManager 19
 - BOTTOM command 176
 - browse user PDS 134
- ## C
- C 78
 - Candle Electronic Customer Support (CECS) 483
 - CANDLLIB
 - in extract JCL 358
 - catalog
 - updating 234
 - catalog, shadow
 - advantage of extract over 335
 - vs. real catalog 234
 - caution
 - when running extract JCL 354
 - caution using force extract run 376
 - CEXPL command
 - restriction in batch 418
 - checkpoints 396
 - Classic Interface - !DB/EXPLAIN panel 195
 - CLIST for OMEGAMON II for DB2 104
 - CLIST, executing 128, 172
 - CLISTs
 - determining extract ID used 337
 - COBOL 78
 - command, JCL 355
 - commands
 - and &variable 322
 - and MOD option 325
 - BOTTOM 176
 - CEXPL
 - restrictions on use 418
 - controlling output destination 319
 - DO commands 130
 - documentation conventions 16
 - DOS command 143
 - DOWN 176
 - END 175
 - FORMAT 170
 - FORMAT? command 167
 - global commands 129
 - JCL? 407
 - to create batch jobs 420
 - PDEBUGOFF 474

- commands (*continued*)
 - PDEBUGON 474
 - RETURN 175
 - SORT 144–146
 - SORTD 144
 - SPLIT 174
 - to browse user PDS 134
 - to edit user PDS 134
 - TOP 176
 - TSO
 - executing 128, 172
 - UP 176
- commands, filtering 155
- commands, trace 470
- comments
 - in batch input commands 430, 443
 - in macros 185
- commit points, frequency of 479
- concurrent access 477
 - batch and online 477
 - overriding default values
 - input to KTECNTL job 482
 - variables affecting 477
 - recommendations 480
 - recommendations for batch environment 482
 - recommendations for online environment 481
 - setting defaults 480
- concurrent batch and online sessions 418
- concurrent extracts
 - restriction 353
- control area (CA), VSAM 390
- control interval (CI), VSAM 390
- control members
 - shipped with !DB/EXPLAIN 405
- conventions, documentation 15
 - braces 16
 - brackets 16
 - data set names 16
 - identifiers 16
 - italics 16
 - qualifiers 16
 - related to figures 16
 - related to panels 16
- convert extract 359, 377
- convert run, extract 377
- cost
 - analyzing by DBRM 265
 - analyzing by package 264
 - cost analysis
 - printing results of 264
 - creating another ISPF session 174
 - creating batch jobs
 - JCL? command 407
 - CUA options
 - customer support 461, 483
 - customizing !DB/EXPLAIN 301
 - customizing extract ID 346
 - customizing the extract JCL 357
 - customizing the extract job 357
- D**
 - DASD
 - estimating 382
 - for non-VSAM files 386
 - for VSAM clusters 383
 - dataset names
 - specifying multiple using KTEON 122
 - specifying using KTEON 120
 - dataset size 377
 - datasets
 - specifying for !DB/EXPLAIN extract 397
 - day
 - on PURGE TIMESTAMP 366
 - DB2
 - propagating !DB/EXPLAIN to 343
 - DB2 ID
 - relationship to extract ID 336
 - DB2, distributed
 - batch functions 235
 - default server 228
 - defining on SET Defaults panel 228
 - defining with SETL command 228
 - specifying 228
 - usage scenarios 233
 - DB2, remote
 - specifying 228
 - usage scenarios 233
 - DB2,remote
 - batch functions 235
 - DB2idUTIL
 - creating JCL for 440
 - !DB/EXPLAIN
 - and MAINT command 458
 - batch facilities 399
 - customizing 301
 - example macros shipped with 188

- !DB/EXPLAIN (*continued*)
 - maintenance level 458
 - propagating to additional subsystems 343
 - scenarios for use 237, 240
 - system facilities 181
 - usage scenarios 237, 240
 - using system facilities 181
- !DB/EXPLAIN CLIST 73
- !DB/EXPLAIN, accessing
 - and displaying statements 101
 - by invoking !DB/EXPLAIN CLIST 73
 - from an edit macro 77
 - from KTEON 90
 - from KTEXPL 78
 - from KTEXPLA 92
 - from KTEXPLB 86
 - from KTEXPLBA 95
 - from KTEXPLD 98
 - on first execution 76
- !DB/EXPLAIN, initiating
 - from KTEXPL 81
- DBRM file
 - estimating space for 386
- DBRMs
 - analyzing cost 265
- !DB/Tools
 - and MAINT? command 458
 - maintenance level 458
- debugging
 - additional information 467
 - and interactive trace 471
 - extract 378
 - flow 378
 - interactive statement execution 467
 - I/O 378
 - parser 378, 474
 - parser interpretation 467
 - structures 378
- debugging, parser
 - batch 475
 - online 474
- DEBUGON/DEBUGOFF command 472
- default filtering 150
- defining an extract ID 342
 - on the initial extract job 342
- definition
 - extract 334
 - extract ID 336
- destination, output
 - default 319, 320
 - for commands and selects 319
 - Output Options Defaults panel 319, 320
 - Output Options panel 319, 320
- developing applications
 - performing remote EXPLAINS 224
- diagram
 - extract syntax 371
- diagrams 371, 372
 - extract syntax 371
 - purge syntax 372
- direct invocation of clists 90
- disk space
 - for non-VSAM files 386
- disk storage
 - estimating 382
- display
 - terminology: vs. panel 17
- displaying extract history 380
- distributed DB2
 - batch functions 235
 - !DB/EXPLAIN prerequisites 218
 - !DB/EXPLAIN support for 217
 - gathering catalog statistics 220
 - initiating remote EXPLAIN 220
 - overriding specification of 227
 - prerequisites for using 218
 - remote functions with !DB/EXPLAIN 220
 - specifying remote DB2 228
 - subsystem location 218
 - updating remote catalog 220
 - usage scenarios 233
- distributed EXPLAIN
 - commands and selects for 221
- DO command 130
- documentation conventions 15
 - % 16
 - braces 16
 - brackets 16
 - data set names 16
 - for commands 16
 - function keys 16
 - identifiers 16
 - italics 16
 - panel vs. display 17
 - qualifiers 16
 - related to figures 16
 - related to panels 16
 - revision bars 16

- documentation conventions (*continued*)
 - terminology 17
 - documentation set 18
 - documentation, online 19
 - DOS command 143
 - DOWN command 176
 - DROP
 - on MERGE SINCE extract function option 362
 - DSNLIB
 - in extract job 358
 - dummy extract 359
 - dummy extract run 377
 - dummy run
 - extract 377
 - to calculate space 377
 - to size datasets 377
 - dynamic dataset allocation
 - overriding in extract job 358
 - dynamic SQL
 - accessing from Primary Menu 109
- E**
- edit macro 78
 - edit macros
 - determining extract ID used 337
 - edit user PDS 134
 - effects of filtering from function to function 159
 - effects of filtering from panel to panel 159
 - electronic customer support 483
 - elements of Administration Menu 110
 - elements of JCL Generation Menu 112
 - elements of Primary Menu 108
 - END command 175
 - entities
 - extracted
 - definition 403
 - extracted, definition 411
 - nonextracted
 - definition 403
 - nonextracted, definition 411
 - used application development 411
 - entities, nonextracted
 - remote EXPLAINS for 224
 - error messages, online 179
 - errors, recovering from 140
 - estimating DASD 382
 - estimating space
 - for non-VSAM files 386
 - for VSAM clusters 383
 - example extract JCL 375
 - example tasks 237, 240
 - EXECs
 - determining extract ID used 337
 - EXECS to access !DB/EXPLAIN 90
 - execution and tracing, statement 471
 - execution, statement
 - and interactive trace 471
 - exiting !DB/EXPLAIN 69, 106
 - EXPLAIN history, saving 375
 - EXPLAIN, remote
 - commands and selects for 221
 - for application development 224
 - initiating 220
 - saving results 232
 - EXPLAIN, remote batch 235
 - EXPLAINing SQL code
 - EXPLAINS, remote
 - for developing applications 224
 - for nonextracted entities 224
 - external sorts 396
 - extract
 - actions to take 353
 - advantage over shadow catalog 335
 - considerations 353
 - data 334
 - debugging 378
 - definition of 334
 - displaying history 380
 - dummy 377
 - examples 375
 - executing 354
 - features of 334
 - function options 360
 - history 380
 - JCL for executing 354, 355
 - main discussion 333
 - masking 348
 - 'Merge Since'
 - access 266
 - change to 266
 - execute 266
 - executing 266
 - planning for 352
 - preparing for executing 355
 - running 354

- extract (*continued*)
 - specifying function options 360
 - specifying permanent storage 397
 - specifying work datasets 397
 - things to do 353
 - using history 380
 - viewing history 380
- extract data
 - definition of 334
- extract debugging 378
- extract function option table 361
- extract function options
 - ALIAS 364
 - MERGE 361
 - MERGE SINCE 362
 - MERGE SINCE DROP 362
 - MERGE SINCE TIMESTAMP 362
 - optional 363, 364, 365
 - REPLACE 361
 - required 361, 362
 - SYNONYM 365
 - THRESHOLD 363
- extract function options, specifying 360
- extract history
 - misuse of REPLACE with 380
 - printing 380
 - saving generations of 380
 - specifying generations of 380
 - viewing 380
- extract ID
 - adding, procedure for 342
 - customizing 346
 - defining on initial extract job 342
 - definition of 336
 - features of 336
 - multiple 344
 - procedure for adding 342
 - relationship to DB2 ID 336
 - relationship to MVS ID 336
 - relationship to shared DASD 336
 - selecting 341
 - single 344
 - used in online session 337
 - viewing, procedure for 338
- extract JCL
 - CANDLLIB in job stream 358
 - caution when running 354
 - customizing 357
 - !DB/EXPLAIN message library allocation 358
- extract JCL (*continued*)
 - DSNLIB in job stream 358
 - KTEMMSG in job stream 358
 - load library allocation 358
 - preparing for executing 355
 - syntax 356
 - SYSCATLG in job stream 358
 - SYSPLAN in job stream 358
 - SYSSTMT in job stream 358
- extract job 355
 - CANDLLIB in 358
 - customizing 357
 - !DB/EXPLAIN message library allocation 358
 - DSNLIB in 358
 - KTEMMSG in 358
 - load library allocation 358
 - overriding dynamic dataset allocation 358
 - SYSCATLG allocation 358
 - SYSPLAN allocation 358
 - SYSSTMT allocation 358
- extract masking 348
- extract run
 - convert 359, 377
 - dummy 359, 377
 - force 359, 376
 - normal 359
 - purge 359
 - purpose 359
 - restart 359, 376
 - specifying purpose 359
- extract summary table 367
- extract syntax diagram 371
- extract, force
 - caution using 376
- extract, normal
 - function options 361
 - normal run function options 361
 - specifying function options 361
- extract, purge
 - function options 366
 - purge run function options 366
 - specifying function options 366
- extract, restart 376
- extracted entities
 - definition 403, 411
- extracts
 - concurrent
 - restriction on 353
 - restriction on concurrent 353

F

facilities, batch 399
 features
 changes 21
 new 21
 features of the extract 334
 features of the extract ID 336
 filtering 147
 and additional sessions 159
 contrast with masking 149
 default 150
 determining when in effect 148
 effects from function to function 159
 effects from panel to panel 159
 input panel 152
 option codes 156
 procedure for defining default 150
 purpose 148
 setting criteria 151
 setting up a filtering macro 160
 subsetting objects to display 155
 wild cards used in 156
 Filtering commands 155
 filtering input panel 152
 filtering, when in effect 148
 filters
 on the command line 151
 resetting 158
 use of 147
 flow debugging 378
 flow trace switch 379
 force extract 359
 force extract run 376
 force run
 caution about 376
 force run, extract 376
 FORMAT command 170
 FORMAT? command 167
 Fortran 78
 forward slash (/)
 in batch input commands 443
 in macros 185
 fragmentation, VSAM 390
 frequency of commit points 479
 function keys
 documentation conventions 16
 function keys, displaying assignments 173

function options
 purge 366
 function options, extract 360
 function options, normal extract 361
 function options, purge extract 366
 functions, basic 67
 functions, distributed 220
 functions, remote 220
 fundamentals 67

G

gather catalog statistics
 remote 226
 GENERATION purge function option 366
 getting in !DB/EXPLAIN 69
 global command 129

H

Help, online
 displaying 179
 history records 396
 history, extract
 displaying 380
 misuse of REPLACE with 380
 saving generations of 380
 specifying generations of 380
 using 380
 history, saving 375
 Housekeeping Menu
 access from Administration Menu 111
 Housekeeping variables
 for concurrent access 477

I

IDCAMS
 performance comparison 394
 initiating a batch job 413
 invoking !DB/EXPLAIN 69
 I/O debugging 378
 I/O trace switch 379

J

JCL

- creating

- to run DB2idUTIL 440

- to run KTEBUTIL 440

- to run the batch utility 440

- extract 355

- JCL command 355

- JCL Generation Menu

- accessing 112

- elements of 112

- JCL Menu

- accessing from Primary Menu 109

- JCL, extract 355

- caution when running 354

- JCL? command 420

- used to create batch jobs 407

K

- keyrange

- implications of using 392

- KSDS structure 389

- KTC CLIST for OMEGAMON II for DB2 104

- KTCIJXTE 404

- KTCTRACE 472

- KTE 73

- KTEBDBR2 404

- KTEBDBRM 404

- KTEBSQLS 404

- KTEBUTIL 404

- creating JCL for 440

- KTECNTL 405

- KTEMSG

- in extract job 358

- KTEON

- accessing !DB/EXPLAIN from 90

- background about using 90

- functions of 119

- selecting object to EXPLAIN 119

- specifying dataset names 120

- specifying multiple dataset names 122

- specifying patterns for dataset names 124

- using member lists with 123

- KTEON EXEC Menu Panel 91

- KTEPFVAR 405

- KTEQMF 87

- determining extract ID used 337

- setting up command synonym 88

- KTEQMFGE 87

- KTESYSIN 405

- KTESYSIS 405

- KTEXPL

- accessing !DB/EXPLAIN from 78, 81

- determining extract ID used 337

- initiating an !DB/EXPLAIN 81

- used for application development 78

- KTEXPLA

- accessing !DB/EXPLAIN from 92

- determining extract ID used 337

- direct invocation 90

- directly invoking 90

- KTEXPLB 404

- accessing !DB/EXPLAIN from 86

- determining extract ID used 337

- KTEXPLBA

- accessing !DB/EXPLAIN from 95

- direct invocation 90

- directly invoking 90

- KTEXPLD

- accessing !DB/EXPLAIN from 98

- determining extract ID used 337

- direct invocation 90

- directly invoking 90

L

- licensing requirements 219

M

- MACEX1 188

- MACEX2 188

- MACEX3 188

- MACEX4 188

- MACEX5 189

- MACEX6 189

- macro facility 183

- macro, edit 78

- macros

- adding a comment 185

- applying a subsequent command to each

- object 185

- batch examples 188

- macros (*continued*)
 - continuing a line to the next line 185
 - examples shipped with !DB/EXPLAIN 188
 - online examples 188
 - suspending processing 185
 - macros, example
 - online 188
 - MAINT command 458
 - MAINT? command 458
 - maintenance level
 - determining 458
 - MAINT command 458
 - MAINT? command 458
 - many extract IDs
 - advantages 344
 - defining 344
 - disadvantages 344
 - masking 348
 - contrast with filtering 149
 - indication of 148
 - indicator of 348
 - package 374
 - plan 374
 - restriction with sessions 118
 - syntax for 349
 - masking, batch
 - separate specification for plans and packages 348
 - masking, extract 348
 - masking, online 348
 - masks
 - restriction on changing defaults 118
 - MERGE extract example 375
 - MERGE extract function option 361
 - MERGE SINCE DROP extract function option 362
 - MERGE SINCE extract example 375
 - MERGE SINCE extract function option 362
 - Merge Since Extract panel 266
 - MERGE SINCE TIMESTAMP extract example 375
 - MERGE SINCE TIMESTAMP extract function option 362
 - messages 459
 - messages, online 179
 - migrating plan table data 374
 - migration
 - and historical plan table 374
 - MOD option
 - on Output Options Defaults panel 319, 325
 - multiple extract IDs
 - advantages 344
 - defining 344
 - disadvantages 344
 - MVS ID
 - relationship to extract ID 336
- N**
- nonextracted entities
 - definition 403, 411
 - nonextracted entity
 - commands and selects
 - for remote EXPLAIN 225
 - performing remote EXPLAIN 224
 - non-VSAM files
 - estimating space for 386
 - normal extract 359
 - normal extract function options, specifying 361
 - normal extract run 361
- O**
- obsolete profile variables 414
 - OMEGAMON II for DB2 104
 - OMEGAMON II for DB2
 - accessing !DB/Tools from 104
 - online documentation 19
 - online example macros 188
 - online Help
 - resolving problems 459
 - online masking 348
 - Online Menu
 - functions of 119
 - selecting object to EXPLAIN 119
 - specifying dataset names 120
 - specifying multiple dataset names 122
 - specifying patterns for dataset names 124
 - using member lists with 123
 - online session
 - concurrent access with batch 477
 - variables affecting 477
 - determining extract ID for 337
 - online sessions
 - concurrent with batch sessions 418

Index

- opcodes 156
 - filtering 156
- optional function option keywords 363
- options for batch jobs 413
- options, output
 - default 320
 - Output Options Defaults panel 320
 - Output Options panel 320
- output destination
 - default 319, 320
 - for commands and selects 319
 - Output Options Defaults panel 319, 320
 - Output Options panel 319, 320
- output options
 - default 320
 - Output Options Defaults panel 320
 - Output Options panel 320
- Output Options Defaults panel
 - &CNT variable 319, 322
 - MOD option 319, 325
- P**
- PACK
 - estimating space for 386, 387
- package masking 374
 - syntax for 349
- package specification 374
 - masking 349
 - syntax diagram 371, 372
- package statement file
 - estimating space for 386, 387
- packages
 - analyzing cost 264
- panel
 - terminology: vs. display 17
- PANELID command 174
- Panels
 - KTEON EXEC Menu Panel 91
 - Merge Since Extract 266
 - Options menu 195
 - Sample Extract 266
- parser
 - debugging 474
 - PDEBUGOFF command 474
 - PDEBUGON command 474
- parser debugging 378
 - in batch 475
 - online 474
 - parser trace switch 379
- PDEBUGOFF command 474
- PDEBUGON command 474
- performance
 - BUILD 392
 - related to PLAN_TABLE organization 392
- PFSHOW command 173
- plan masking 374
 - syntax for 349
- plan specification 374
 - example 373
 - masking 349
 - on SYSIN input 348
 - package specification
 - syntax diagram 371, 372
- plan statement file
 - estimating space for 386
- plan table data
 - historical 374
 - merging from different users 374
- PLAN_TABLE
 - extracting from user-defined 375
 - merging information from different users 375
- PLAN_TABLE Support 375
- PL/I 78
- portable document format, Adobe 14
- Primary Menu
 - accessing 108
 - elements of 108
 - returning to 175
- printing extract history 380
- problem resolution 455
 - additional information 467
 - interactive statement execution 467
 - parser interpretation 467
- profile dataset
 - customizing extract ID in 346
 - use to tailor !DB/EXPLAIN 346
- profile variable input, building
 - PROFOUTE 417
- Profile variables
 - controlling overrides in batch 414
 - for batch jobs 413
 - obsolete 414
 - Overriding 405
- PROFOUTE 417
- prompted queries
 - EXPLAINing 87

- PURGE extract example 375
- purge extract function options, specifying 366
- purge extract run 366
- purge function options
 - GENERATION 366
 - SYNC 366
- PURGE processing 366
- purge summary table 367
- purge syntax diagram 372
- purpose of extract run 359

- Q**
- QLOC 156
- QMF 78
 - German language bridge 87
 - setting up command synonym 88
- QMF bridge 87
- queries, EXPLAINing
 - QMF prompted queries 87
 - QMF QBE queries 87
 - QMF SQL queries 87
- query-by-example queries
 - EXPLAINing 87
- ?, in Select field 137
- quick access
 - applying selects to multiple objects 138

- R**
- rebind, remote 227
- recovery 395
- recovery from errors 140
- REFRESH
 - restriction on use 118
- REFRESH command
 - restriction when in an !DB/EXPLAIN session 118
- remote batch functions 235
- remote bind/rebind
 - commands to perform 227
- remote catalog statistics
 - commands to gather 226
 - commands to update 226
- remote DB2
 - batch functions 235
 - specifying 228
 - usage scenarios 233
- remote EXPLAIN
 - commands and selects for 221
 - for application development 224
 - saving results 232
- remote EXPLAIN, batch 235
- remote EXPLAIN, initiating 220
- remote EXPLAINs
 - for developing applications 224
 - for nonextracted entities 224
- remote functions 220
- remote processing
 - gathering catalog statistics 220
 - updating remote catalog 220
- remote server
 - default 228
 - on SET Defaults panel 228
 - using SETL command 228
- remote subsystem
 - overriding specification of 227
- REORG considerations 390
- REPLACE extract example 375
- REPLACE extract function option 361
- requirements, licensing 219
- resetting filters 158
- resolving problems
 - additional information 467
 - interactive statement execution 467
 - parser interpretation 467
- resources, system
 - effect of sessions on 115
- RESTART 396
 - specifying permanent storage 397
 - specifying work datasets 397
- restart extract 359
- restart extract run 376
- restart run
 - restrictions on 376
- restart run, extract 376
- restriction
 - on changing mask defaults 118
 - on concurrent extracts 353, 477
 - on use of REFRESH command 118
- restrictions
 - for working with sessions 118
 - on session switching 118
 - when changing mask values 118
 - when using REFRESH command 118
- RETURN command 175

Index

- REUSE option (VSAM) 392
 - revision bars 16
 - run, convert extract 377
 - run, dummy extract 377
 - run, force extract 376
 - run, normal extract 361
 - run, purge extract 366
 - run, restart extract 376
- S**
- Sample Extract panel 266
 - scenario
 - print cost analysis 264
 - scrolling
 - Number of lines, altering 178
 - to the beginning of a document or list 177
 - to the end of a document 177
 - to the end of a list 177
 - selecting extract ID 341
 - selects
 - and &CNT variable 322
 - and MOD option 325
 - controlling output destination 319
 - fast path for 138
 - using on object list panel 138
 - selects, using 135
 - server, remote
 - specifying 228
 - session switching
 - restriction on 118
 - sessions
 - beginning 116
 - effect on system resources 115
 - ending 117
 - restriction on changing mask values 118
 - restrictions on 118
 - switching 116
 - Sessions Menu
 - accessing 116
 - background about 114
 - elements of 114
 - Setting up a Filtering Macro 160
 - shadow catalog
 - advantage of extract over 335
 - vs. real catalog 234
 - single extract ID
 - advantages 344
 - defining 344
 - single extract ID (*continued*)
 - disadvantages 344
 - size, dataset 377
 - solving problems
 - additional information 467
 - interactive statement execution 467
 - parser interpretation 467
 - SORT command 144–146
 - SORTD command 144
 - sorted data
 - returning to original order 146
 - sorts, external 396
 - space
 - refining estimates 384
 - space calculation 377
 - space, estimating 382
 - for non-VSAM files 386
 - for VSAM clusters 383
 - specifying extract function options 360
 - specifying normal extract function options 361
 - specifying purge extract function options 366
 - SPLIT command 174
 - split screen in ISPF 174
 - splits, VSAM 390
 - SPUFI 78
 - SQL queries
 - EXPLAINing 87
 - SQL, dynamic
 - accessing from Primary Menu 109
 - starting !DB/EXPLAIN 69
 - statement execution, tracing 471
 - statements, displaying
 - when accessing !DB/EXPLAIN 101
 - STMT
 - estimating space for 386
 - storage, permanent
 - specifying for !DB/EXPLAIN extract 397
 - structures debugging 378
 - structures trace switch 379
 - submitting a batch job 412
 - switch
 - flow trace 379
 - I/O trace 379
 - parser trace 379
 - structures trace 379
 - switching, session
 - restriction on 118
 - SYNC purge function option 366

SYNONYM extract function option 365
 syntax
 extract JCL 356
 extract job 356
 package masking 349
 plan masking 349
 timestamp masking 349
 syntax diagram
 extract 371
 purge 372
 SYSCATLG
 allocation for extract job 358
 in extract job 358
 SYSPLAN
 allocation for extract job 358
 in extract job 358
 SYSSTMT
 allocation for extract job 358
 in extract job 358
 system maintenance
 level of 458
 version 458
 system resources
 effect of sessions on 115

T

table, extract function option 361
 table, extract summary 367
 table, purge summary 367
 tasks, example 237, 240
 telephone support 484
 terminology
 [] 17
 { } 17
 documentation conventions 17
 panel vs. display 17
 THRESHOLD extract function option 363
 time
 on PURGE TIMESTAMP 366
 TIMESTAMP
 on MERGE SINCE extract function
 option 362
 on PURGE processing 366
 timestamp masking
 syntax for 349
 TOP command 176

trace commands 470
 trace facility 472
 trace, interactive
 and statement execution 471
 tracing
 commands for 470
 tracing statement execution 471
 troubleshooting 455
 TSO commands, executing 128, 172

U

UBROWSE command 134
 UEDIT command 134
 underscore 156
 UP command 176
 update catalog statistics
 remote 226
 user PDS
 command to browse 134
 command to edit 134
 user-defined PLAN_TABLE 375
 merging information from different users 375
 using selects 135
 Using the OMEGAMON II for DB2 Callable
 Interface
 Classic Interface - !DB/EXPLAIN panel 195
 utility, batch
 creating JCL for 440

V

variable, profile
 building input
 PROFOUTE 417
 variables, profile
 controlling overrides in batch 414
 for batch jobs 413
 obsolete 414
 viewing extract history 380
 viewing specific extract ID 338
 VSAM
 control area (CA) 390
 control interval (CI) 390
 estimating space for clusters 383
 fragmentation 390
 KSDS structure 389
 LISTCAT 390

Index

VSAM (*continued*)
 REUSE option 390
VSAM clusters
 planning for 388

W

wild cards 156
 filtering 156

User Comment Form

!DB®/EXPLAIN for DB2 User's Guide Version 500

TE54-5842-3

Please take a moment to share your comments and suggestions regarding Candle's documentation. Be as specific as possible.

Page #	Comment
①	_____
②	_____
③	_____
④	_____
⑤	_____
⑥	_____
⑦	_____

Please provide the following information. Thank you for your time and assistance.

Name

Job Function

Company

Site ID

Address

You may return your comments to us at your convenience. Upon receipt, we will respond to you within five business days.



FAX
(310) 727-4131



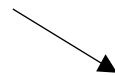
Fold, tape, and mail
(postage paid)
Please do not staple.

May we call to discuss your comments?

YES Phone number: _____ Best time to call: _____

NO

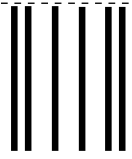
Cut or fold
along line



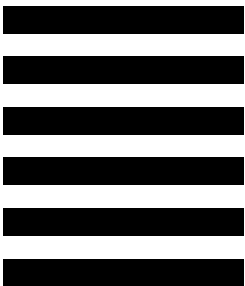
Fold and tape

Please do not staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO 559 SANTA MONICA CA

POSTAGE WILL BE PAID BY ADDRESSEE



ATTN INFORMATION DEVELOPMENT
CANDLE CORPORATION
201 N DOUGLAS STREET
EL SEGUNDO CA 90245



Fold and tape

Please do not staple

Fold and tape



Cut or fold
along line



* T E 5 4 - 5 8 4 2 - 3 *