

DB2 for OS/390
Version 5



Installation Guide

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page ix.

First Edition (June 1997)

This edition applies to Version 5 of IBM DATABASE 2 Server for OS/390 (DB2 for OS/390), 5655-DB2, and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

The technical changes for this edition are summarized under “Summary of Changes to this Book” in the Introduction. Specific changes are indicated by a vertical bar to the left of a change. A vertical bar to the left of a figure caption indicates that the figure has changed. Editorial changes that have no technical significance are not noted.

This softcopy version is based on the printed edition of the book and includes the changes indicated in the printed version by vertical bars. Additional changes made to this softcopy version of the manual since the hardcopy manual was published are indicated by the hash (#) symbol in the left-hand margin.

© Copyright International Business Machines Corporation 1982, 1997. All rights reserved.

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

Contents

Notices	ix
Programming Interface Information	ix
Trademarks	xi

Section 1. Introduction 1-1

Chapter 1-1. Introduction to This Book and the DB2 Library	1-3
Who Should Read This Book	1-3
How This Book Is Organized	1-3
How to Read the Syntax Diagrams	1-4
How to Use the DB2 Library	1-5
How to Obtain DB2 Information	1-7
DB2 Classes	1-9
Summary of Changes to DB2 for OS/390 Version 5	1-9
Summary of Changes to This Book	1-16
Chapter 1-2. System Planning Concepts	1-19
The Relational Database	1-19
Structured Query Language (SQL)	1-20
The Structure of DB2	1-20
Control and Maintenance of DB2	1-31
DB2 and the MVS Environment	1-33
Data Sharing	1-43

Section 2. Planning and Installing DB2 2-1

Chapter 2-1. Introduction to Installation and Migration	2-9
Features	2-9
Installation and Migration Steps Overview	2-10
SMP/E Steps Summary	2-11
Installation Steps Summary	2-11
Migration Steps Summary	2-12
Fallback Steps Summary	2-13
Remigration Steps Summary	2-14
Chapter 2-2. Estimating DB2 Storage Needs	2-15
DASD Storage for DB2 Subsystem	2-15
DASD Storage for User Data	2-25
Virtual Storage for Address Spaces	2-34
Virtual Storage for Storage Pools and Working Storage	2-38
Real Storage	2-48
Chapter 2-3. Setting Up DB2 Online Help	2-51
Installing the Online Help Reader	2-52
Unloading DB2 Books	2-59
Verifying Online Help and Setting Exit Options	2-63
Using Online Help	2-65
Maintaining Online Help	2-67

	Chapter 2-4. Loading DB2 Libraries	2-69
	What IBM Sends You	2-69
	What You Produce	2-70
	SMP/E Step 1: Copy and Edit the SMP/E Jobs	2-72
	SMP/E Step 2: Allocate Distribution and Target Libraries: DSNTIJAE	2-80
	SMP/E Step 3: Run the RECEIVE Job: DSNTIJRC	2-81
	SMP/E Step 4: Cleanup Job for Migration: DSNTIJUD (Optional)	2-82
	SMP/E Step 5: Run the APPLY Job: DSNTIJAP	2-83
	SMP/E Step 6: Run the ACCEPT Job: DSNTIJAC	2-84
	SMP/E Step 7: Unload the Jobs for the DB2I Panels	2-84
	SMP/E Step 8: Allocate Libraries for DB2I Panels	2-85
	SMP/E Step 9: Run the RECEIVE Job for the DB2I Panels	2-85
	SMP/E Step 10: Run the APPLY Job for the DB2I Panels	2-85
	SMP/E Step 11: Run the ACCEPT Job for the DB2I Panels	2-86
#	SMP/E Step 12: Copy and Edit the SMP/E Jobs for DB2 REXX Language	
#	Support	2-86
#	SMP/E Step 13: Run the RECEIVE Job for DB2 REXX Language Support	2-86
#	SMP/E Step 14: Run the APPLY Job for DB2 REXX Language Support	2-86
#	SMP/E Step 15: Run the ACCEPT Job for DB2 REXX Language Support	2-86
	SMP/E Step 16: Ensure Installation of Proper Maintenance	2-86
	Finishing SMP/E Processing	2-87
	 Chapter 2-5. Installing, Migrating, and Updating System Parameters	2-89
	Running the Installation CLIST	2-89
	Data Parameters Panel: DSNTIPA2	2-105
	Define Group or Member Panel: DSNTIPK	2-110
	System Resource Data Set Names: DSNTIPH	2-112
	Data Set Names Panel 1: DSNTIPT	2-116
	Data Set Names Panel 2: DSNTIPU	2-121
	Data Set Names Panel 3: DSNTIPQ	2-127
	Data Set Names Panel 4: DSNTIPG	2-130
	Data Set Names Panel 5: DSNTIPW	2-133
	CICS Release Panel: DSNTIPV	2-135
	Install DB2 CICS Datasets Panel: DSNTIP3	2-136
	Sizes Panel: DSNTIPD	2-139
	Thread Management Panel: DSNTIPE	2-145
	Buffer pool Sizes Panel 1: DSNTIP1	2-150
	Buffer pool Sizes Panel 2: DSNTIP2	2-152
	Tracing Panel: DSNTIPN	2-154
	Operator Functions Panel: DSNTIPO	2-158
	Application Programming Defaults Panel 1: DSNTIPF	2-164
	Application Programming Defaults Panel 2: DSNTIP4	2-170
	IRLM Panel 1: DSNTIPI	2-174
	IRLM Panel 2: DSNTIPJ	2-177
	Protection Panel: DSNTIPP	2-181
	MVS PARMLIB Updates Panel: DSNTIPM	2-186
	Active Log Data Set Parameters: DSNTIPL	2-190
	Archive Log Data Set Parameters: DSNTIPA	2-193
	Databases and Spaces to Start Automatically Panel: DSNTIPS	2-199
	Distributed Data Facility: DSNTIPR	2-201
	Distributed Data Facility Panel: DSNTIP5	2-205
	Stored Procedures Parameters Panel: DSNTIPX	2-207
	Data Definition Control Support Panel: DSNTIPZ	2-209
	Job Editing Panel: DSNTIPY	2-212

Install DB2—CLIST Calculations Panel 1: DSNTIPC	2-215
Install DB2—CLIST Calculations Panel 2: DSNTIPC1	2-219
Completing the CLIST Processing	2-219
The Update Process	2-228
Chapter 2-6. Installing the DB2 Subsystem	2-233
Installation Step 1: Define DB2 to MVS: DSNTIJMV	2-233
Installation Step 2: Define the Integrated Catalog Facility Catalog and Alias: DSNTIJCA	2-238
Installation Step 3: Define System Data Sets: DSNTIJIN	2-239
Installation Step 4: Initialize System Data Sets: DSNTIJID	2-239
Installation Step 5: Define DB2 Initialization Parameters: DSNTIJUZ	2-240
Installation Step 6: Define User Authorization Exit Routines: DSNTIJEX (Optional)	2-241
Installation Step 7: Record DB2 Data to SMF (Optional)	2-242
Installation Step 8: Establish Subsystem Security (Optional)	2-243
Installation Step 9: Connect DB2 to TSO	2-243
Installation Step 10: Connect IMS to DB2 (Optional)	2-250
Installation Step 11: Connect CICS to DB2 (Optional)	2-250
Installation Step 12: IPL MVS	2-251
Installation Step 13: Start the DB2 Subsystem	2-251
Installation Step 14: Define Temporary Work Files: DSNTIJTM	2-252
Installation Step 15: Define and Bind DB2 Objects and User-Maintained Databases: DSNTIJSG	2-254
Installation Step 16: Populate the User-Maintained Databases (Optional)	2-255
Installation Step 17: Bind the Packages for DB2 REXX Language Support: DSNTIJRX	2-256
Installation Step 18: Back Up the DB2 Directory and Catalog: DSNTIJIC	2-256
Installation Step 19: Verify the Installation	2-257
Installing a Communications Network	2-257
Installing a Second DB2 on the Same Operating System	2-257
Considerations for Using Shared DASD	2-263
Enabling stored procedures after installation	2-263
Chapter 2-7. Migrating the DB2 Subsystem	2-267
Migration Considerations	2-267
Migration Step 1: Run Link Checker on DB2 for MVS/ESA Version 4 Table Spaces (optional)	2-272
Migration Step 2: Determine Which Plans and Packages Are Invalid after Migration (Optional)	2-273
Migration Step 3: Check for Consistency between Catalog Tables (Optional)	2-274
Migration Step 4: Image Copy Directory and Catalog in Case of Fallback: DSNTIJIC	2-274
Migration Step 5: Connect DB2 to TSO	2-275
Migration Step 6: Connect IMS to DB2 (Optional)	2-278
Migration Step 7: Connect CICS to DB2 (Optional)	2-278
Migration Step 8: Stop DB2 for MVS/ESA Version 4 Activity	2-278
Migration Step 9: Back up Your DB2 for MVS/ESA Version 4 Volumes (Optional)	2-280
Migration Step 10: Define DB2 Initialization Parameter: DSNTIJUZ	2-280
Migration Step 11: Establish Subsystem Security (Optional)	2-281
Migration Step 12: Define DB2 for OS/390 Version 5 to MVS: DSNTIJMV	2-281
Migration Step 13: Define System Data Sets: DSNTIJIN	2-284
Migration Step 14: Define User Authorization Exit Routines: DSNTIJEX	2-284

#

Migration Step 15: IPL MVS	2-285
Migration Step 16: Start DB2 for OS/390 Version 5	2-285
Migration Step 17: Tailor DB2 for OS/390 Version 5 Catalog: DSNTIJTC	2-286
Migration Step 18: Ensure There Are No Problems with the Catalog (Optional)	2-287
Migration Step 19: Prepare Dynamic SQL Program: DSNTIJTM	2-287
Migration Step 20: Bind DB2 objects and User Maintained Database Activity: DSNTIJSG	2-287
Migration Step 21: Bind the Packages for DB2 REXX Language Support: DSNTIJRX	2-288
Migration Step 22: Image Copy DB2 for OS/390 Version 5 Catalog: DSNTIJIC	2-289
Migration Step 23: Verify Your DB2 for OS/390 Version 5 System	2-289
Migration Step 24: Make Adjustments for Release Incompatibilities	2-290
Migration Step 25: Convert Catalog and Directory Indexes (Optional)	2-294
Chapter 2-8. Falling Back and Remigrating	2-297
Falling Back	2-297
Remigrating	2-303
Chapter 2-9. Verifying with the Sample Applications	2-305
Installation Verification Phases and Programs	2-306
Planning for Verification	2-308
Special Considerations for COBOL Programs	2-308
Special Considerations for C and C++ Programs	2-311
Special Considerations for PL/I Programs	2-312
Phase 0: Deleting the Sample Objects (DSNTEJ0)	2-312
Phase 1: Creating and Loading Sample Tables	2-313
Phase 2: Testing the Batch Environment	2-316
Phase 3: Testing SPUFI, DRDA Access, Dynamic SQL, and TSO	2-319
Phase 4: Testing the IMS Environment	2-323
Phase 5: Testing the CICS Environment	2-327
Phase 6: Accessing Data at a Remote Site	2-331
The Sample Applications	2-341
Scenarios	2-344
Edit Exit Routine	2-363
Huffman Compression Exit Routine	2-363
Sample Field Procedure	2-363
Dynamic SQL Statements (DSNTEA, DSNTEA, DSNTEA)	2-363
Dynamic SQL Programs (DSNTIAD, DSNTEP2, DSNTEAUL)	2-366
Chapter 2-10. Connecting the IMS Attachment Facility	2-371
Making DB2 Load Modules Available to IMS	2-371
Defining DB2 to IMS	2-372
IMS Attachment Facility Macro (DSNMAPN)	2-377
Chapter 2-11. Connecting the CICS Attachment Facility	2-379
Using the Attachment Facility for CICS Version 4 and Subsequent Releases	2-379
Using the Attachment Facility for CICS Version 3	2-380
Calculating Space Requirements for the CICS Attachment Facility	2-381
Defining the DB2 to CICS Connection Using the RCT	2-385
Updating the CICS System Tables	2-385
Updating CICS Initialization JCL	2-396
Coordinating DB2 and CICS Security	2-396

Preparing CICS Applications for DB2	2-397
CICS Attachment Facility Macro (DSNCRCT)	2-398

Section 3. Communicating with Other Systems 3-1

Chapter 3-1. Connecting Distributed Database Systems	3-3
Installing the DB2 Subsystem	3-4
How the Communications Database (CDB) Handles Requests	3-5
Support for Asynchronous Transfer Mode (ATM) Networks	3-6
DB2 and the DCE Directory	3-6

Chapter 3-2. Connecting Systems with VTAM	3-9
Step 1: Install VTAM Version 3.4.2 or Version 4.1	3-11
Step 2: Choose Names and a Password	3-11
Step 3: Define the DB2 Subsystem to VTAM	3-13
Step 4: Populate the Communications Database	3-20
Step 5: Start VTAM to Use DB2	3-24
Step 6: Tune the System	3-24
Calculating Session Limits	3-33
Calculating VTAM I/O Buffer Pool (IOBUF) Storage	3-36
Interpreting CNOS Messages	3-37
Sample VTAM Definitions to Connect Two DB2s	3-39
Using the Change Log Inventory Utility to Update the BSDS	3-46

Chapter 3-3. Connecting Systems with TCP/IP	3-49
Enabling TCP/IP communication	3-50
Step 1: Install OpenEdition MVS and TCP/IP and Prepare the LE/370 Runtime Library	3-52
Step 2: Enable DDF for OpenEdition	3-52
Step 3: Define the DB2 Subsystem to TCP/IP	3-53
Step 4: Populate the Communications Database	3-56
Step 5: Start TCP/IP support	3-58
Step 6: Tuning TCP/IP	3-59

Appendixes X-1

Appendix A. Sample Application Output	X-3
Phase 1 (DSNTEJ1P Job)	X-3
Phase 2	X-11
Phase 3 (DSNTEJ3P Job)	X-31
Phases 4 and 5	X-53

Appendix B. Character Conversion	X-55
Understanding Character Conversion	X-55
Specifying a System Coded Character Set Identifier	X-56
Customizing Character Conversion	X-62
When Remote Packages Should be Rebound	X-64

Glossary and Bibliography G-1

Glossary	G-3
---------------------------	------------

Bibliography G-21

Index I-1

Index I-3

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in
this document. The furnishing of this document does not give you any license to
these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Licensees of this program who wish to have information about it for the purpose of
enabling (1) the exchange of information between independently created programs
and other programs (including this one) and (2) the mutual use of the information
that has been exchanged, should contact:

IBM Corporation
IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Programming Interface Information

This book is intended to help you to install IBM DATABASE 2 Server for OS/390 (DB2 for OS/390).

This book also documents General-use Programming Interface and Associated Guidance Information provided by IBM DATABASE 2 Server for OS/390.

General-use programming interfaces allow the customer to write programs that obtain the services of DB2 for OS/390.

General-use Programming Interface and Associated Guidance Information is identified where it occurs, by an introductory statement to a chapter or section or by the following markings:

_____ General-use Programming Interface _____

General-use Programming Interface and Associated Guidance Information

_____ End of General-use Programming Interface _____

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

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs either by an introductory statement to a chapter or section or by the following marking:

_____ Product-sensitive Programming Interface _____

Product-sensitive Programming Interface and Associated Guidance Information

_____ End of Product-sensitive Programming Interface _____

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

3090	ES/9000
AD/Cycle	GDDM
APL2	Hiperspace
BookManager	IBM
C/370	IBMLink
C/MVS	IMS
CBPDO	IMS/ESA
CBIPO	Language Environment
CICS	MVS/DFP
CICS/ESA	MVS/ESA
CICS/MVS	MVS/SP
COBOL/370	MVS/XA
DataPropagator	NetView
DataHub	OpenEdition
Distributed Relational Database Architecture	OS/2
DATABASE 2	OS/390
DB2	OS/400
DB2/2	Parallel Sysplex
DB2/6000	PR/SM
DFSMS	QMF
DFSMS/MVS	RACF
DFSMSdfp	RAMAC
DFSMSdss	System/370
DFSMShsm	System/390
DFSORT	Systems Application Architecture
DProp	SAA
DRDA	SQL/DS
DXT	VTAM

Throughout the library, the DB2 licensed program and a particular DB2 subsystem are each referred to as “DB2.” In each case, the context makes the meaning clear. The term *MVS* is used to represent the MVS/Enterprise Systems Architecture (MVS/ESA); *CICS* is used to represent CICS/MVS and CICS/ESA; *IMS* is used to represent IMS/ESA; *C* and *C language* are used to represent the C/370 and C/C++ for MVS/ESA programming languages. COBOL is used to represent OS/VS COBOL, VS COBOL II, IBM COBOL, and COBOL/370 programming languages.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Section 1. Introduction

Chapter 1-1. Introduction to This Book and the DB2 Library	1-3
Who Should Read This Book	1-3
How This Book Is Organized	1-3
How to Read the Syntax Diagrams	1-4
How to Use the DB2 Library	1-5
How to Obtain DB2 Information	1-7
DB2 on the Web	1-7
DB2 Publications	1-7
How to Order the DB2 Library	1-8
DB2 Classes	1-9
Summary of Changes to DB2 for OS/390 Version 5	1-9
Server Solution	1-9
Performance	1-11
Increased Capacity	1-13
Improved Availability	1-13
Client/Server and Open Systems	1-13
User Productivity	1-15
Summary of Changes to This Book	1-16
Section 1. System Planning Concepts	1-16
Section 2. Planning and Installing DB2	1-17
Section 3. Communicating with Other Systems	1-17
Section 4. Appendix	1-17
Chapter 1-2. System Planning Concepts	1-19
The Relational Database	1-19
Structured Query Language (SQL)	1-20
The Structure of DB2	1-20
Data Structures	1-21
System Structures	1-26
Control and Maintenance of DB2	1-31
Commands	1-31
Utility Jobs	1-31
High Availability	1-32
DB2 and the MVS Environment	1-33
Address Spaces	1-34
DB2 and MVS	1-35
DB2 and RACF	1-36
DB2 and SMS	1-36
DB2 and TSO Attachment Facility	1-36
DB2 and ISPF	1-37
Call Attachment Facility	1-38
DB2 and CICS	1-38
DB2 and IMS	1-40
DB2 and DL/I Batch	1-42
DB2 and DDF	1-42
Data Sharing	1-43

Chapter 1-1. Introduction to This Book and the DB2 Library

This chapter contains specific information about this book and a general overview of the DB2 library.

Who Should Read This Book

This book is primarily intended for those responsible for installing DB2 or setting up DB2 for distributed communications. This book is intended for those who plan to install DB2 from the host, using the installation CLIST. If you plan to install DB2 from your workstation, using DB2 Installer see *DB2 Installer User's Guide*.

This book assumes that the reader is familiar with:

- The basic concepts and facilities of DB2
- The MVS Time Sharing Option (TSO) and the MVS Interactive System Productivity Facility (ISPF)
- The basic concepts of Structured Query Language (SQL)
- The basic concepts of Customer Information Control System (CICS)
- The basic concepts of Information Management System (IMS)
- How to define and allocate MVS data sets using MVS job control language (JCL)
- How to use the IBM System Modification Program/Extended (SMP/E) to install IBM licensed programs.

To set up DB2 for distributed communications, knowledge of Virtual Telecommunications Access Method (VTAM) or Transmission Control Protocol/Internet Protocol (TCP/IP) is needed also.

How This Book Is Organized

This book has the following sections.

- Section 1. Introduction

This section introduces the DB2 system and explains the concepts that relate to installing DB2 and setting up DB2 for distributed communications.

- Section 2. Planning and Installing DB2

This section provides information on estimating DB2 storage requirements, explains how to choose parameter values, gives step-by-step installation and migration instructions, and shows how to verify installation and migration.

- Section 3. Communicating with Other Systems

This section describes the steps you take to connect DB2 subsystems for communication using the distributed data facility (DDF).

- Appendix A. Sample Application Output

This contains parts of the sample output from the jobs in Phases 1, 2, and 3 of installation verification.

- Appendix B. Character Conversion for Distributed Data

This describes how DB2 handles character conversion for distributed data.

How to Read the Syntax Diagrams

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

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

The \blacktriangleright — symbol indicates the beginning of a statement.

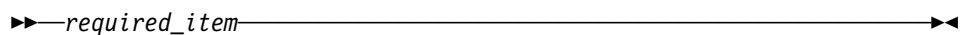
The — \blacktriangleright symbol indicates that the statement syntax is continued on the next line.

The \blacktriangleright — symbol indicates that a statement is continued from the previous line.

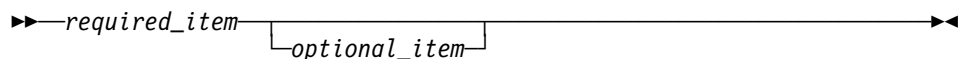
The — \blacktriangleleft symbol indicates the end of a statement.

Diagrams of syntactical units other than complete statements start with the \blacktriangleright — symbol and end with the — \blacktriangleright symbol.

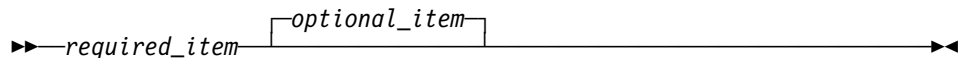
- Required items appear on the horizontal line (the main path).



- Optional items appear below the main path.

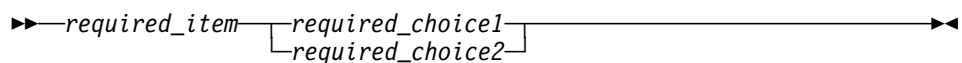


If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.

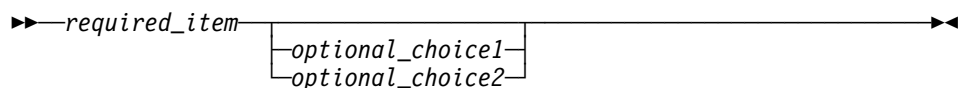


- If you can choose from two or more items, they appear vertically, in a stack.

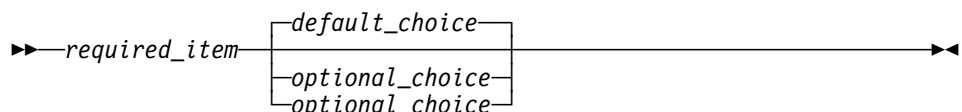
If you *must* choose one of the items, one item of the stack appears on the main path.



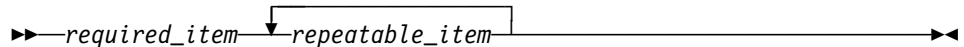
If choosing one of the items is optional, the entire stack appears below the main path.



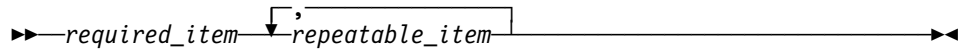
If one of the items is the default, it appears above the main path and the remaining choices are shown below.



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



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



A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Keywords appear in uppercase (for example, FROM). They must be spelled exactly as shown. Variables appear in all lowercase letters (for example, *column-name*). They represent user-supplied names or values.
- If punctuation marks, parentheses, arithmetic operators, or other such symbols are shown, you must enter them as part of the syntax.

How to Use the DB2 Library

Titles of books in the library begin with DB2 for OS/390 Version 5. However, references from one book in the library to another are shortened and do not include the product name, version, and release. Instead, they point directly to the section that holds the information. For a complete list of books in the library, and the sections in each book, see the bibliography at the back of this book.

Throughout the library, the DB2 for OS/390 licensed program and a particular DB2 for MVS/ESA subsystem are each referred to as “DB2.” In each case, the context makes the meaning clear.

The most rewarding task associated with a database management system is asking questions of it and getting answers, the task called *end use*. Other tasks are also necessary—defining the parameters of the system, putting the data in place, and so on. The tasks associated with DB2 are grouped into the following major categories (but supplemental information relating to all of the below tasks for new releases of DB2 can be found in *Release Guide*):

Installation: If you are involved with DB2 only to install the system, this book might be all you need.

If you will be using data sharing then you also need *Data Sharing: Planning and Administration*, which describes installation considerations for data sharing.

End use: End users issue SQL statements to retrieve data. They can also insert, update, or delete data, with SQL statements. They might need an introduction to SQL, detailed instructions for using SPUFI, and an alphabetized reference to the types of SQL statements. This information is found in *Application Programming and SQL Guide* and *SQL Reference*.

End users can also issue SQL statements through the Query Management Facility (QMF) or some other program, and the library for that program might provide all the

instruction or reference material they need. For a list of some of the titles in the QMF library, see the bibliography at the end of this book.

Application Programming: Some users access DB2 without knowing it, using programs that contain SQL statements. DB2 application programmers write those programs. Because they write SQL statements, they need *Application Programming and SQL Guide*, *SQL Reference*, and *Call Level Interface Guide and Reference* just as end users do.

Application programmers also need instructions on many other topics:

- How to transfer data between DB2 and a host program—written in COBOL, C, or FORTRAN, for example
- How to prepare to compile a program that embeds SQL statements
- How to process data from two systems simultaneously, say DB2 and IMS or DB2 and CICS
- How to write distributed applications across platforms
- How to write applications that use DB2 Call Level Interface to access DB2 servers
- How to write applications that use Open Database Connectivity (ODBC) to access DB2 servers
- How to write applications in the Java programming language to access DB2 servers

The material needed for writing a host program containing SQL is in *Application Programming and SQL Guide* and *Application Programming Guide and Reference for Java*. The material needed for writing applications that use DB2 Call Level Interface or ODBC to access DB2 servers is in *Call Level Interface Guide and Reference*.

For handling errors, see *Messages and Codes*.

Information about writing applications across platforms can be found in *Distributed Relational Database Architecture: Application Programming Guide*.

System and Database Administration: *Administration* covers almost everything else. *Administration Guide* divides those tasks among the following sections:

- Section 2 (Volume 1) of *Administration Guide* discusses the decisions that must be made when designing a database and tells how to bring the design into being by creating DB2 objects, loading data, and adjusting to changes.
- Section 3 (Volume 1) of *Administration Guide* describes ways of controlling access to the DB2 system and to data within DB2, to audit aspects of DB2 usage, and to answer other security and auditing concerns.
- Section 4 (Volume 1) of *Administration Guide* describes the steps in normal day-to-day operation and discusses the steps one should take to prepare for recovery in the event of some failure.
- Section 5 (Volume 2) of *Administration Guide* explains how to monitor the performance of the DB2 system and its parts. It also lists things that can be done to make some parts run faster.

In addition, the appendixes in *Administration Guide* contain valuable information on DB2 sample tables, National Language Support (NLS), writing exit routines, interpreting DB2 trace output, and character conversion for distributed data.

If you are involved with DB2 only to design the database, or plan operational procedures, you need *Administration Guide*. If you also want to carry out your own plans by creating DB2 objects, granting privileges, running utility jobs, and so on, then you also need:

- *SQL Reference*, which describes the SQL statements you use to create, alter, and drop objects and grant and revoke privileges
- *Utility Guide and Reference*, which explains how to run utilities
- *Command Reference*, which explains how to run commands

If you will be using data sharing, then you need *Data Sharing: Planning and Administration*, which describes how to plan for and implement data sharing.

Additional information about system and database administration can be found in *Messages and Codes*, which lists messages and codes issued by DB2, with explanations and suggested responses.

Diagnosis: Diagnosticians detect and describe errors in the DB2 program. They might also recommend or apply a remedy. The documentation for this task is in *Diagnosis Guide and Reference* and *Messages and Codes*.

How to Obtain DB2 Information

DB2 on the Web

Stay current with the latest information about DB2. View the DB2 home page on the World Wide Web. News items keep you informed about the latest enhancements to the product. Product announcements, press releases, fact sheets, and technical articles help you plan your database management strategy. Technical professionals can access DB2 publications on the Web and follow links to other Web sites with more information about DB2 family and OS/390 solutions. Access DB2 on the Web with the following URL:

<http://www.ibm.com/software/db2os390>

DB2 Publications

The DB2 publications are available in both hardcopy and softcopy format. Using online books on CD-ROM, you can read, search across books, print portions of the text, and make notes in these BookManager books. With the appropriate BookManager READ product or IBM Library Readers, you can view these books on the MVS, VM, OS/2, DOS, AIX and Windows platforms.

When you order DB2 Version 5, you are entitled to one copy of the following CD-ROM, which contains the DB2 licensed book for no additional charge:

DB2 Server for OS/390 Version 5 Licensed Online Book, LK2T-9075.

You can order multiple copies for an additional charge by specifying feature code 8207.

When you order DB2 Version 5, you are entitled to one copy of the following CD-ROM, which contains the DB2 and DATABASE 2 Performance Monitor online books for no additional charge:

DB2 Server for OS/390 Version 5 Online Library, SK2T-9092

You can order multiple copies for an additional charge through IBM's publication ordering service.

Periodic updates will be provided on the following collection kit available to licensees of DB2 Version 5:

IBM Online Library Transaction Processing and Data Collection, SK2T-0730

SK2T-9092 will be superseded by SK2T-0730 when updates to the online library are available.

In some countries, including the United States and Canada, you receive one copy of the collection kit at no additional charge when you order DB2 Version 5. You will automatically receive one copy of the collection kit each time it is updated, for no additional charge. To order multiple copies of SK2T-0730 for an additional charge, see "How to Order the DB2 Library." In other countries, updates will be available in displayable softcopy format in the IBM Online Book Library Offering (5636-PUB), SK2T-0730 IBM Online Library Transaction Processing and Data Collection at a later date.

See your IBM representative for assistance in ordering the collection.

DB2 Server for OS/390 books are also available for an additional charge on the following collection kits, which contain online books for many IBM products:

IBM Online Library MVS Collection, SK2T-0710, in English

Online Library Omnibus Edition OS/390 Collection, SK2T-6700, in English

IBM Online Library MVS Collection Kit, SK88-8002, in Japanese, for viewing on DOS and Windows platforms

How to Order the DB2 Library

You can order DB2 publications and CD-ROMs through your IBM representative or the IBM branch office serving your locality. If you are located within the United States or Canada, you can place your order by calling one of the toll-free numbers :

- In the U.S., call 1-800-879-2755.
- In Canada, call 1-800-565-1234.

To order additional copies of licensed publications, specify the SOFTWARE option. To order additional publications or CD-ROMs, specify the PUBLICATIONS & SLSS option. Be prepared to give your customer number, the product number, and the feature code(s) or order numbers you want.

DB2 Classes

IBM Education and Training offers a wide variety of classroom courses to help you quickly and efficiently gain DB2 expertise. Classes are scheduled in cities all over the world. For more information, including the current local schedule, please contact your IBM representative.

Classes can also be taught at your location, at a time that suits your needs. Courses can even be customized to meet your exact requirements. The diagrams below show the DB2 curriculum in the United States. *Enterprise Systems Training Solutions, GR28-5467* describes these courses. You can inquire about or enroll in them by calling 1-800-IBM-TEACH (1-800-426-8322).

Figure 1. Application Programmer Curriculum

Figure 2. Application Designer Curriculum

Figure 3. Database Administrator Curriculum

Figure 4. System Administrator Curriculum

Figure 5. System Programmer Curriculum

Figure 6. Migration Curriculum

Summary of Changes to DB2 for OS/390 Version 5

DB2 for OS/390 Version 5 delivers a database server solution for OS/390. Version 5 supports all functions available in DB2 for MVS/ESA Version 4 plus enhancements in the areas of performance, capacity, and availability, client/server and open systems, and user productivity.

If you are currently using DB2, **you can migrate only from a DB2 for MVS/ESA Version 4 subsystem**. This summary gives you an overview of the differences to be found between these versions.

Server Solution

OS/390 retains the classic strengths of the traditional MVS/ESA operating system, while offering a network-ready, integrated operational environment.

The following features work directly with DB2 for OS/390 applications to help you use the full potential of your DB2 subsystem:

- Net.Data for OS/390
- DB2 Installer
- DB2 Estimator for Windows
- DB2 Visual Explain

- Workstation-based Performance Analysis and Tuning
- DATABASE 2 Performance Monitor

Net.Data for OS/390

Net.Data provides support for Internet access to DB2 data through a Web server. Applications built with Net.Data make data stored in any DB2 server more accessible and useful. Net.Data Web applications provide continuous application availability, scalability, security, and high performance.

This no charge feature can be ordered with DB2 Version 5 or downloaded from Internet. The Net.Data URL is:

<http://www.ibm.com/software/data/net.data/downloads.html>

DB2 Installer

DB2 Installer offers the option to install DB2 on an OS/2 workstation. Now, you can use a friendly graphical interface to complete installation tasks easily with DB2 Installer.

This function is delivered on CD-ROM with DB2 Visual Explain.

DB2 Estimator for Windows

DB2 Estimator provides an easy-to-use capacity planning tool. You can estimate the sizes of tables and indexes, and the performance of SQL statements, groups of SQL statements (transactions), utility runs, and groups of transactions (capacity runs). From a simple table sizing to a detailed performance analysis of an entire DB2 application, DB2 Estimator saves time and lowers costs. You can investigate the impact of new or modified applications on your production system, *before* you implement them.

This no charge feature can be ordered with DB2 Version 5 or downloaded from the Internet. From the internet, use the IBM Software URL:

<http://www.ibm.com/software/>

From here, you can access information about DB2 Estimator using the download function.

DB2 Visual Explain

DB2 Visual Explain lets you tune DB2 SQL statements on an OS/2 workstation. You can see DB2 EXPLAIN output in a friendly graphical interface and easily access, modify, and analyze applications with DB2 Visual Explain.

Workstation-based Performance Analysis and Tuning

The new workstation-based Performance Analysis and Tuning function simplifies system administration. You can access statistical data to help you analyze and improve system performance. This function works with the optional DB2 PM feature to provide full analysis and tuning functionality.

DATABASE 2 Performance Monitor (DB2 PM)

DB2 PM lets you monitor, analyze, and optimize the performance of DB2 Version 5 and its applications. An online monitor, for both host and workstation environments, provides an immediate "snap-shot" view of DB2 activities and allows for exception processing while the system is operational. The workstation-based online monitor can connect directly to the Visual Explain function of the DB2 base product.

DB2 PM also offers a history facility, a wide variety of customizable reports for in-depth performance analysis, and an EXPLAIN function to analyze and optimize SQL statements. For more information, see *DB2 PM for OS/390 General Information*.

This feature can be ordered with DB2 Version 5.

Performance

Sysplex Query Parallelism

The increased power of Sysplex query parallelism in DB2 for OS/390 Version 5 allows DB2 to go far beyond DB2 for MVS/ESA Version 4 capabilities; from the ability to split and process a single query within a DB2 subsystem to processing that same query across many different DB2 subsystems in a data sharing group.

The advances this release offers in scalable query processing let you process queries quickly while accommodating the potential growth of data sharing groups and the increasing complexity of queries.

Prepared Statement Caching

DB2 reduces the cost of duplicate prepares for the same dynamic SQL statement by saving them in a cache. Now, different application processes can share prepared statements and they are preserved past the commit point. This performance improvement offers the most benefit for:

- Client/server applications that frequently use dynamic SQL for repeated execution of SQL statements
- Relatively short dynamic SQL statements for which PREPARE cost accounts for most of the CPU expended

Reoptimization

When host variables, parameter markers, or special registers were used in previous releases, DB2 could not always determine the best access path because the values for these variables were unknown. Now, you can tell DB2 to reevaluate the access path at run time, after these values are known. As a result, queries can be processed more efficiently, and response time is improved.

Faster Transactions and Batch

- Caching of package authorization improves performance at run time for remote packages and applications that use pattern-matching characters in a package list.
- You can define a table space to use ***selective partition locking***, which can reduce locking costs for applications that do partition-at-a-time processing. It also can reduce locking costs for certain data sharing applications that rely on an affinity between members and data partitions.

- A new standalone utility lets you preformat active logs.
- With LOAD and REORG, you can preformat data sets up to the high allocated RBA, which can make processing for sequential inserts more predictable.

Faster Utilities

- LOAD and REORG jobs run faster and more efficiently with enhanced index key sorting that reduces CPU and elapsed time, and an inline copy feature that lets you make an image copy without a separate copy step.
- New REORG options let you select rows to discard during a REORG and, optionally, write the discarded records to a file.
- When you run the REBUILD, RECOVER, REORG, or LOAD utility on DB2-managed indexes or table spaces, a new option lets you logically reset and reuse the DB2-managed objects.
- RECOVER INDEX and LOAD run faster on large numbers of rows per page.
- Sampling support for RUNSTATS reduces the processing required to collect nonindexed column statistics.
- BSAM striping improves the I/O capability of DB2 utilities.

Other Performance Enhancements

- There are several significant performance enhancements to data sharing, including selective partition locking, the MAXROWS option, and several optimizations to reduce data sharing overhead.
- DB2 installations that run in the OS/390 Version 2 Release 6 environment can now have as many as (approximately) 25 000 open DB2 data sets at one time. The maximum number of open data sets in earlier releases of OS/390 is 10000.
- You can easily alter the length of variable-length character columns using the new ALTER COLUMN clause of the ALTER TABLE statement.
- SQL CASE expressions let you eliminate queries with multiple UNIONS and improve performance by using only one table scan.
- You can collect a new statistic on concatenated index keys to improve the performance of queries with correlated columns. The statistic lets DB2 estimate the number of rows that qualify for the query more accurately, and select access paths more efficiently.
- DB2 scans partitions more efficiently and allows scans during parallel processing.
- Query enhancements include the ability to:
 - Use indexes for joins on string columns that have different lengths
 - Use an index to access predicates with noncorrelated IN subqueries
- Noncolumn expressions in simple predicates are evaluated at stage 1 and can be indexable.

Increased Capacity

DB2 for OS/390 Version 5 introduces the concept of a *large* partitioned table space. Defining your table space as large allows a substantial capacity increase: to approximately one terabyte of data and up to 254 partitions. In addition to accommodating growth potential, large partitioned table spaces make database design more flexible, and can improve availability.

Improved Availability

Online REORG

DB2 for OS/390 Version 5 adds a major improvement to availability with *Online REORG*. Now, you can avoid the severe availability problems that occurred while offline reorganization of table spaces restricted access to read only during the unload phase and no access during reload phase of the REORG utility. Online REORG gives you full read and write access to your data through most phases of the process with only very brief periods of read only or no access.

Data Sharing Enhancements

- Version 5 provides continuous availability with group buffer pool duplexing. Prior releases of DB2 rely on DASD and the merged recovery logs to recover group buffer pool (GBP) data that is lost if a coupling facility fails. With group buffer pool duplexing, DB2 writes changed pages to both a *primary GBP* and a *secondary GBP*. Overlapped writes to the GBPs provide good performance and eliminate the writes to DASD.
- Group buffer pool rebuild makes coupling facility maintenance easier and improves access to the group buffer pool during connectivity losses.
- Automatic group buffer pool recovery accelerates GBP recovery time, eliminates operator intervention, and makes data available faster when GBPs are lost because of coupling facility failures.
- Improved restart performance for members of a data sharing group reduces the impact of retained locks by making data available faster when a group member fails.
- Changes to traces and DISPLAY GROUPBUFFERPOOL output improve monitoring.

Tracker site for disaster recovery

You can set up a tracker site that shadows the activity of a primary site, and eliminate the need to constantly ship image copies.

Client/Server and Open Systems

Native TCP/IP Network Support

DB2's support of TCP/IP networks allows DRDA clients to connect directly to DDF and eliminate the gateway machine. In addition, customers can now use asynchronous transfer mode (ATM) as the underlying communication protocol for both SNA and TCP/IP connections to DB2.

Stored Procedures

- Return multiple SQL result sets to local and remote clients in a single network operation.
- Receive calls from applications that use standard interfaces, such as Open Database Connectivity** (ODBC) and X/Open** Call Level Interface, to access data in DB2 for OS/390.
- Run in an enhanced environment. DB2 supports multiple stored procedures address spaces managed by the MVS Workload Manager (WLM). The WLM environment offers efficient program management and allows WLM-managed stored procedures to run as subprograms and use RACF security.
- Use individual MVS dispatching priorities to improve stored procedure scheduling.
- Access data sources outside DB2 with two-phase commit coordination.
- Use an automatic COMMIT feature on return to the caller that reduces network traffic and the length of time locks are held.
- Have the ability to invoke utilities, which means you can now invoke utilities from an application that uses the SQL CALL statement.
- Support IMS Open Database Access (ODBA). Now a DB2 stored procedure can directly connect to IMS DBCTL and access IMS data.

Dynamic Query and Network Performance

Improvements for DRDA Applications

- Reduced processing costs for block fetch operations
- DRDA support for OPTIMIZE FOR n ROWS on SELECT
- Faster dynamic SQL queries and reduced processing costs for VTAM network operations
- Reduced message traffic for dynamic SQL SELECT statements

Improved Application Portability

- DB2 for OS/390 Version 5 introduces the DB2 Call Level Interface (CLI) to MVS/ESA. Unlike applications that use embedded SQL to access DB2 data, applications that choose CLI are not tied to a precompiler, packages, or a plan.

Workstation and desktop applications use standard interfaces, such as Open Database Connectivity (ODBC), to access relational data. Standard interfaces need one version of an application to access many data sources. Now, you can port UNIX workstation and PC desktop applications to DB2 for OS/390 and exploit the CLI (ODBC) capabilities without modification. In addition, applications can issue ODBC or CLI calls from within a stored procedure.
- You can now access DB2 for OS/390 databases in your Java applications. You can use DB2 Connect Java Database Connectivity (JDBC) for your dynamic SQL applications, or SQLJ for your static SQL applications.
- DB2 adds DRDA support for the DESCRIBE INPUT statement to improve performance for many ODBC applications.
- Now, you can write multithreaded DB2 CLI applications, and restrictions on connection switching no longer exist.

- DB2 now provides ASCII table support for clients and servers across platforms. This support reduces the cost of translation between EBCDIC and ASCII encoding schemes. ASCII table support also offers an alternative to writing field procedures that provide the ASCII sort sequence, which improves performance.

Improved Security

- DB2 for OS/390 supports Distributed Computing Environment (DCE) for authenticating remote DRDA clients. DCE offers the following benefits:
 - Network security: By providing an encrypted DCE ticket for authentication, remote clients do not need to send an MVS password in readable text.
 - Simplified security administration: End users do not need to maintain a valid password on MVS to access DB2; instead, they maintain their DCE password only.
- New descriptive error codes help you determine the cause of network security errors.
- You can change end user MVS passwords from DRDA clients.

User Productivity

Improved SQL Compatibility

DB2 conforms to the ANSI/ISO SQL entry level standard of 1992. Application programmers can take advantage of a more complete set of standard SQL to use across the DB2 family to write portable applications. New SQL function includes:

- More check options for view definitions.
- Foreign keys that reference UNIQUE keys as well as PRIMARY keys.
- An extension to GRANT that lets the REFERENCES privilege apply to a list of columns.
- A new delete rule, NO ACTION, that you can use to define referential constraints for self-referencing tables.
- SQL CASE expressions provide the capability to create conditional logic wherever an expression is allowed.
- SQL temporary tables allow application programs to easily create and use temporary tables that store results of SQL transactions without logging or recovery.

New Access Choice

A new attachment facility, the Recoverable Resource Manager Services attachment facility, improves access in a client/server environment. It coordinates two-phase commit processing between DB2 and other participating resource managers in any MVS application environment. Other key features include the ability for multiple users to run in a single address space, thread reuse, and moving threads between MVS tasks.

Image Copy Enhancements

The COPY, LOAD, and REORG utilities provide:

- Features of the COPY utility that help you quickly determine what type of image copy to take, when to take it, and let DB2 automatically take it for you.
- Inline copy in LOAD and REORG that lets you create an image copy while improving data availability.

Improved Integration of C++ and IBM COBOL for MVS & VM Support

It is easier for application programmers to use object-oriented programming techniques in their DB2 applications. DB2 for OS/390 Version 5 adds COBOL and C++ languages as options on installation panels, DB2I panels, the DSNH command, and DCLGEN.

Other Usability Enhancements

- To prevent long running units of work and to help avoid unnecessary work during the recovery phase of restart, DB2 issues new warning messages at an interval of your choice.
- A new special register for decimal precision provides better granularity, so that applications that need different values for decimal precision can run in the same DB2 subsystem.
- Trace records for IFCID 0022 now include most information in the PLAN_TABLE.
- An increase from 127 to 255 rows on a page improves table space processing and eliminates the need for compression.
- Install SYSOPR can recover objects using the START DATABASE command.
- A filtering capability for DISPLAY BUFFERPOOL limits statistics information to a specified set of page sets.
- You can enter comments within the SYSIN input stream for DB2 utilities.

Summary of Changes to This Book

If you plan to install or migrate DB2 using the new workstation installation tool, refer to *DB2 Installer*.

Section 1. System Planning Concepts

This section has the following changes:

- An enhancement allowing online reorg substantially increases availability.
- An updated description of the DB2 catalog.
- The bootstrap dataset (BSDS) now includes a table of IP addresses.
- The communications database (CDB) is redesigned and placed in the system catalog.
- A description of the workload manager-established stored procedures address space.

Section 2. Planning and Installing DB2

This section has the following changes:

- Chapter 2-1. Introduction to Installation and Migration contains updated summaries of installation and migration steps.
- Chapter 2-2. Estimating DB2 Storage Needs provides calculations to help you estimate storage for the DB2 subsystem.
- Chapter 2-3. Setting Up DB2 Online Help contains information for setting up the DB2 Online Help.
- Chapter 2-4. Loading DB2 Libraries gives you the new names of the DB2 target and distribution libraries for both the DB2 base code and the DB2I panels.
- Chapter 2-5. Installing, Migrating, and Updating System Parameters includes an improved format for field descriptions. There are new panels and fields to support C++, TCP/IP, ASCII, and other enhancements.
- Chapter 2-6. Installing the DB2 Subsystem has only minor changes; the installation steps are relatively unchanged.
- Chapter 2-7. Migrating the DB2 Subsystem describes migration considerations for migrating to Version 5 and the migration process.
- Chapter 2-8. Falling Back and Remigrating describes fallback considerations and procedures for Version 5.
- Chapter 2-9. Verifying with the Sample Applications includes information on verifying massive parallel queries.

Section 3. Communicating with Other Systems

This section has the following changes:

- "Chapter 3-1. Connecting Distributed Database Systems" on page 3-3 describes changes to the Communications Database (CDB).
- "Chapter 3-2. Connecting Systems with VTAM" on page 3-9 is a new chapter that describes how to connect DB2 to DRDA partners with VTAM. This was formerly part of "Chapter 3-1. Connecting Distributed Database Systems" on page 3-3.
- "Chapter 3-3. Connecting Systems with TCP/IP" on page 3-49 is a new chapter that describes how to connect DB2 to DRDA partners with TCP/IP.

Section 4. Appendix

"Sample Application Output" is updated with sample output from jobs in Phases 1, 2, and 3. "Character Conversion for Distributed Data" is updated and some of the formats changed.

Chapter 1-2. System Planning Concepts

This chapter introduces the DB2 system and explains the concepts that relate to system and database administration. It consists of the following sections:

- “The Relational Database” is a broad introduction to DB2.
- “The Structure of DB2” on page 1-20 describes the elements you deal with when using DB2.
- “Control and Maintenance of DB2” on page 1-31 briefly describes commands and utility jobs.
- “DB2 and the MVS Environment” on page 1-33 explains how DB2 operates with certain related IBM products.

General information about DB2 for OS/390 is available from the DB2 for OS/390 World Wide Web page:

<http://www.software.ibm.com/data/db2/os390/>

The Relational Database

DB2 is a *relational database management system*. In a relational database, data is perceived to exist in one or more tables, each containing a specific number of *columns* and a number of unordered *rows*. Each column in a row is related in some way to the other columns. Thinking of the data as a collection of tables gives you an easy way to visualize the stored data and enables you to explain your needs in easy-to-understand terms. Table 1 shows the department table (DSN8510.DEPT) of the sample database. The table contains four columns (DEPTNO, DEPTNAME, MGRNO, and ADMRDEPT) and nine rows.

Table 1. Example of a DB2 Table (Department Table)

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00
B01	PLANNING	000020	A00
C01	INFORMATION CENTER	000030	A00
D01	DEVELOPMENT CENTER		A00
E01	SUPPORT SERVICES	000050	A00
D11	MANUFACTURING SYSTEMS	000060	D01
D21	ADMINISTRATION SYSTEMS	000070	D01
E11	OPERATIONS	000090	E01
E21	SOFTWARE SUPPORT	000100	E01

DB2 accesses data by referring to its content instead of its location or organization in storage. The rows of a relational table have no fixed order. The order of the columns, however, are always the order in which you specified them when defining the table.

A DB2 database involves more than just a collection of tables. It also includes table spaces, storage groups, views, indexes, and other items. These are all collectively referred to as DB2 *structures*.

Sample Tables: The examples in this book are based on the set of tables described in Appendix A (Volume 2) of *Administration Guide*. Those tables are part of the DB2 licensed program and represent data related to the activities of an imaginary computer services company, the Spiffy Computer Services Company.

Structured Query Language (SQL)

The language used to access the data in DB2 tables is called SQL. SQL contains both data definition statements and data manipulation statements. Data definition statements define the structures of the database. Data manipulation statements manipulate (retrieve, insert, delete, and update) data in tables. SQL also performs functions that are neither data definition nor data manipulation, such as granting or revoking authorization to use resources.

You can execute statements written in SQL in the following ways:

- Interactively, using the SQL processor using file input (SPUFI) or Query Management Facility (QMF), another IBM licensed program.
- By embedding them in application programs written in Ada, APL2, assembler, BASIC, C, C++, COBOL, FORTRAN, Java, PL/I, Prolog, or REXX.
- By using a DB2 Call Level Interface application program. For more information about using DB2 CLI, see *Call Level Interface Guide and Reference*.

An important aspect of SQL is that it is nonprocedural; that is, when you use SQL you specify *what* you want done, not *how* to do it. In particular, to access data you need only name the table and column where it can be found; you do not have to describe an access method.

Below is an example of an SQL SELECT statement. It selects (retrieves) the department number (DEPTNO) and department name (DEPTNAME) columns from the department table (DSN8510.DEPT). It retrieves only those rows where the administrating department (ADMRDEPT) is D01.

```
SELECT DEPTNO, DEPTNAME
  FROM DSN8510.DEPT
 WHERE ADMRDEPT = 'D01';
```

You can also use SQL statements to insert new rows into a table, update existing rows, or delete rows.

For an elementary introduction to SQL and detailed instructions for using SPUFI, see Section 2 of *Application Programming and SQL Guide*. For a complete reference to the SQL language, see *SQL Reference*.

The Structure of DB2

The elements that DB2 manages can be divided into two broad categories:

- Data structures, which are accessed under the user's direction and by which the user's data (and some system data) is organized.
- System structures, which are controlled and accessed by DB2.

Databases

In DB2, a *database* is a set of DB2 structures. When you define a DB2 database, you give a name to an eventual collection of tables and associated indexes, as well as to the table spaces in which they reside. A single database, for example, can contain all the data associated with one application or with a group of related applications. Collecting that data into one database allows you to start or stop access to all the data in one operation and grant authorization for access to all the data as a single unit.

If you create a table space or a table and do not specify a database, the table or table space is created in the default database, DSNDB04. The default database is predefined in the installation process; its default buffer pool is BP0, and its default DB2 storage group is SYSDEFLT.

Storage group SYSDEFLT is created when you install DB2. After that, all users have the authority to create table spaces or tables in database DSNDB04. The system administrator can revoke those privileges and grant them only to particular users as necessary.

When you migrate to Version 5, DB2 adopts the default database and default storage group you used in Version 4. You have the same authority for Version 5 as you did in Version 4.

DB2 Storage Groups

A DB2 *storage group* is a set of volumes on direct access storage devices (DASD). The volumes hold the data sets in which tables and indexes are actually stored. The description of a storage group names the group and identifies its volumes and the VSAM catalog that records the data sets.

All volumes of a given storage group must have the same device type. But, as Figure 7 on page 1-21 suggests, parts of a single database can be stored in different storage groups.

DFSMS storage groups are also discussed in this book. They are not the same as DB2 storage groups. When DFSMS storage groups are discussed, the reference will be explicit.

Table Spaces

A *table space* is one or more data sets in which one or more tables are stored. A table space can consist of a number of VSAM data sets, which can together contain up to 64 gigabytes of data for tablespaces not defined as large. A LARGE table space can consist of up to 254 data sets, or 1 terabyte of data. Data sets are VSAM linear data sets (LDSs). Table spaces are divided into equal-sized units, called *pages*, which are written to or read from DASD in one operation. Refer to the following sections for maximum sizes of each table space type. You should also refer to Appendix A of *SQL Reference* for specific limits.

When you create a table space, you can specify the database to which the table space belongs and the storage group it uses. If you do not specify the database and storage group, DB2 assigns the table space to the default database and the default storage group. You also control whether the table space is *partitioned*, *segmented*, or *simple*.

|

#

#

Partitioned Table Spaces: In a partitioned table space, the available space is divided into separate units of storage called *partitions*, each containing a part of one table. Although the partitions can be independently assigned to separate storage groups, the entire collection of data is logically a single table. A partition for a table space not defined as large can be 1, 2, or 4 gigabytes in length, depending on the number of partitions contained in the entire table space. If less than 16 partitions are defined on the table space, then each partition's maximum size is 4 gigabytes. For a table space that is defined as large, the maximum size of a partition is 4 gigabytes, for 1 to 254 partitions. See for more information on partitioned table spaces.

Partitioning a table space provides several advantages for large tables:

- Improved data availability. You can perform normal maintenance on one partition of the table while the rest of the table remains available for utility or SQL processing.
- Improved utility performance. A utility can work on all partitions simultaneously instead of working on one partition at a time. Also, different utilities can work on different partitions simultaneously. This can significantly reduce the amount of time needed for a utility to finish.
- Improved query response time. When DB2 scans data to answer a query, it scans through partitions simultaneously instead of scanning through the entire table space from beginning to end. This improvement is most significant for queries that are complex or require DB2 to scan a lot of data.
- Improved efficiency in table space and index scans. DB2 can limit a scan to a subset of the partitions in both table space and index scans. Only the specific partitions that are needed are actually scanned.

Segmented Table Spaces: A segmented table space is intended to hold more than one table. The available space is divided into groups of pages called *segments*, each the same size. Each segment contains rows from only one table. To search all the rows for one table, it is not necessary to scan the entire table space, but only the segments that contain that table. If a table is dropped, its segments become immediately reusable. A segmented table space can have between 1 and 32 VSAM linear data sets, all of which are either user-defined or in the same storage group. The maximum size of a data set in the segmented table space is 2 gigabytes. And so, the maximum size of a segmented table space is 64 gigabytes (2 gigabytes multiplied by 32 data sets). See for more information on segmented table spaces.

|
|

Simple Table Spaces: If a table space is not partitioned or segmented, it is called *simple*. A simple table space can contain more than one table, but the rows of different tables are not kept separate. To find all the rows of one table can require scanning the entire table space. If a table is dropped, its rows are not deleted. The space occupied by the rows does not become available until the table space is reorganized. All tables in a table space must be either user-defined or in the same storage group. The maximum size of a data set in the simple table space is 2 gigabytes. And so, the maximum size of a simple table space is 64 gigabytes (2 gigabytes multiplied by 32 data sets).

|
|

#

Tables

All data in a DB2 database is presented in *tables*—collections of rows all having the same columns. When you create a table in DB2, you define an ordered set of columns.

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00
B01	PLANNING	000020	A00
C01	INFORMATION CENTER	000030	A00

A few rows of the sample department table are shown above. The ordered set of columns contains DEPTNO, DEPTNAME, MGRNO, and ADMRDEPT. Each row contains data for a single department; the columns represent, respectively, its number, its name, the employee number of its manager, and the number of the department to which it reports. The whole table is a collection of rows, each containing those columns.

At the intersection of a column and row is a *value*. For example, PLANNING is the value of the DEPTNAME column in the row for Department B01.

The storage representation of a row is called a *record*, and the storage representation of a column is called a *field*. For most tasks discussed in this book, you do not need to know what a record looks like. For instances when you do need to know, see Appendix B (Volume 2) of *Administration Guide*.

All the data in a given column must be of the same data type. For example, the data in the DEPTNAME column of the table is varying-length character (VARCHAR). DB2 data types are described in .

A table can have a *primary key*. A primary key is a column or set of columns whose values uniquely identify each row. In the sample department table, DSN8510.DEPT, the DEPTNO (department ID) column is a primary key. Columns of other tables can be *foreign keys*, whose values must be equal to values of the primary key of the first table. In the sample employee table, DSN8510.EMP (see Appendix A (Volume 2) of *Administration Guide*) the column that shows what department an employee works in is a foreign key; its values must be values of the department ID column in the department table. DB2 can automatically enforce the integrity of references from a foreign key to a primary key by guarding against insertions, updates, or deletions that violate the integrity. Automatic enforcement of referential integrity is described in .

Indexes

An *index* is an ordered set of pointers to the data in a DB2 table. The index is stored separately from the table. Each index is based on the values of data in one or more columns of a table. After you create an index, DB2 maintains the index, but you can check, repair, or reorganize it.

You can create an index on a table any time after you create the table, and you can create the index either before or after you load data into the table. Except for changes in performance, users of the table are unaware that an index is being used. DB2 decides whether or not to use the index to access the table.

#

Each index occupies its own *index space*. A nonpartitioned index on a table space that **is not** defined as *large* can contain from 1 to 32 linear data sets, each data set capable of holding 2 GB of data . A nonpartitioned index on a table space that **is** defined as *large* can contain from 1 to 128 VSAM linear data sets, each capable of holding up to 4 GB of data, for a total of .5 terabytes. When you create an index, an index space is automatically defined in the same database as the table.

Refer to Appendix A of *SQL Reference* for specific limits.

You can use indexes to:

- Improve performance. In many cases, access to data is faster with an index than without.
- Ensure uniqueness. A table with a unique index cannot have two rows with the same values in the column or columns that form the index key.

Both of these topics are discussed further in .

Partitioned Indexes: A partitioned index is created with the keyword PART, on a table in a partitioned table space that is divided into multiple index spaces.

#

A partitioned index consists of between 1 and 64 partitions for a table space that is not defined as large and between 1 and 254 partitions for a table space that is defined as large. Each of these partitions has a one to one correspondence to a VSAM data set. Each partitioned table space always has one table and one partitioned index defined on it. The maximum size of a partitioned index defined on a table space that is not defined as large is 1, 2 or 4 GB depending on the number of partitions. The maximum size of a partitioned data set defined on a table space that is defined as large is 4 GB. A partitioned index is always a clustering index.

Clustering Indexes: A clustering index determines the approximate order in which records of the base table are stored. Therefore, DB2 can access the entire table in the sequence of the clustering key faster than in any other sequence. Each table can have only one clustering index.

Views

A *view* is an alternate way of representing data that exists in one or more tables. A view can include all or some of the columns from one or more base tables. Views can also be based on other views or on a combination of views and tables. Figure 8 shows some of the possible relationships between tables and the views that users see of them.

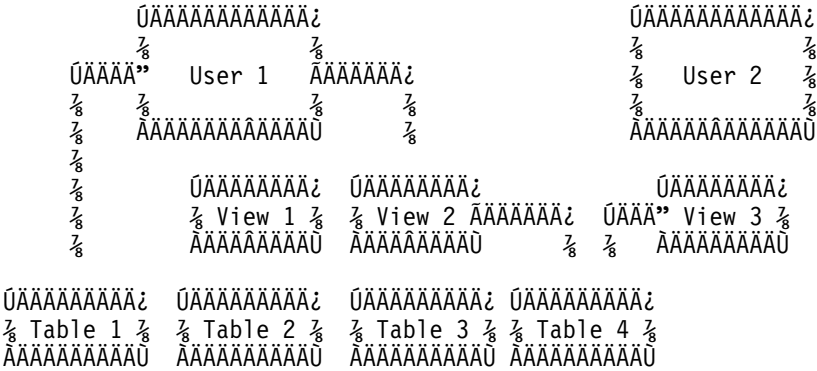


Figure 8. Relationship between Tables and Views

A view looks just like a table and can be used as though it were a table. You can use a view name in the FROM clause of an SQL SELECT statement, in the same way you use a table name. Although some operations cannot be performed on a view, often a user need not know that an apparent table is actually a view.

A table has a storage representation, but a view does not. When you define a view, DB2 stores the definition of the view in the DB2 catalog. But DB2 does not store any data for the view itself, because the data already exists in the base table or tables. Because no data is stored for the view, you cannot create an index on it. However, an index on a base table can improve the performance of operations on the view. For more information on views, see Section 2 of *Administration Guide*.

System Structures

DB2 system structures described in this section include:

- “DB2 Catalog”
- “DB2 Directory” on page 1-27
- “Active and Archive Logs” on page 1-28
- “Bootstrap Data Set (BSDS)” on page 1-29
- “Buffer Pools” on page 1-29
- “Communications Database” on page 1-30
- “Data Definition Control Support Database” on page 1-30
- “Resource Limit Facility Database” on page 1-30

DB2 Catalog

The DB2 catalog consists of tables of data about everything defined to the DB2 system. The DB2 catalog is contained in system database DSNDB06. When you create, alter, or drop any structure, DB2 inserts, updates, or deletes rows of the catalog that describe the structure and tell how the structure relates to other structures. Appendix D of *SQL Reference* describes all the DB2 catalog tables.

For Version 5, the communications database (CDB) is moved into the catalog.

DB2 has extensive support to help move your applications into the next millennium. The Version 5 catalog supports timestamps generated both before and after the year 2000.

To illustrate the use of the catalog, here is a brief description of some of what happens when the employee table is created:

- To record the name of the structure, its owner, its creator, its type (alias, table, or view), the name of its table space, and the name of its database, DB2 inserts a row into the catalog table SYSIBM.SYSTABLES.
- To record the name of the table to which the column belongs, its length, its data type, and its sequence number in the table, DB2 inserts rows into SYSIBM.SYSCOLUMNS for each column of the table.
- To increase by one the number of tables in the table space DSN8S51E, DB2 updates the row in the catalog table SYSIBM.SYSTABLESPACE.
- To record that the owner (DSN8510) of the table has all privileges on the table, DB2 inserts a row into table SYSIBM.SYSTABAUTH.

Because the catalog consists of DB2 tables in a DB2 database, you can use SQL statements to retrieve information from it.

For catalog data set naming conventions,

DB2 Directory

The DB2 directory contains information required to start DB2, and DB2 uses the directory during normal operation. You cannot access the directory using SQL. The structures in the directory are not described in the DB2 catalog.

The directory consists of a set of DB2 tables stored in five table spaces in system database DSNDB01. Each of the following table spaces is contained in a VSAM linear data set:

- SCT02 is the skeleton cursor table space (SKCT).
- SPT01 is the skeleton package table space.
- SYSLGRNX is the log range table space.
- SYSUTILX is the system utilities table space.
- DBD01 is the database descriptor (DBD) table space.

The directory contains DSNNSCT02, an index space for SCT02; DSNNSPT01 and DSNNSPT02, index spaces for SPT01; DSNLLX01 and DSNLLX02, indexes for SYSLGRNX; and DSNLUX01 and DSNLUX02, the indexes for SYSUTILX.

For directory data set naming conventions,

Skeleton Cursor Table: The skeleton cursor table space (SCT02) contains a table that describes the internal form of SQL statements of application programs. When you bind a plan, DB2 creates a skeleton cursor table in SCT02. The index space for the skeleton cursor table is DSNNSCT02.

Skeleton Package Table: The skeleton package table space (SPT01) contains a table that describes the internal form of SQL statements in application programs. When you bind a package, DB2 creates a skeleton package table in SPT01. The index spaces for the skeleton package table are DSNNSPT01 and DSNNSPT02.

Log Range: DB2 inserts a row in the log range table space (SYSLGRNX) every time a table space or partition is opened and updated, and it updates SYSLGRNX whenever that structure is closed. The row contains the opening log relative byte address (RBA), the closing log RBA, or both for the structure. The log RBA is the relative byte address in the log data set where open and close information about the structure is contained.

The use of SYSLGRNX speeds up recovery by limiting the log information that must be scanned to apply changes to a table space or partition being recovered.

The two indexes defined on SYSLGRNX are DSNLLX01 (a clustered index) and DSNLLX02.

System Utilities: DB2 inserts a row in the system utilities table space (SYSUTILX) for every utility job that is run. The row remains there until the utility completes its full processing. If the utility terminates without completing, DB2 uses the information in the row to restart the utility. The indexes defined on SYSUTILX are DSNLUX01 and DSNLUX02.

Database Descriptor: The database descriptor table space (DBD01) contains internal control blocks, called *database descriptors* (DBDs), that describe the databases existing within DB2. Each database has exactly one corresponding DBD

Bootstrap Data Set (BSDS)

The *bootstrap data set* (BSDS) is a VSAM key-sequenced data set (KSDS) that contains information critical to DB2. Specifically, the BSDS contains:

- An inventory of all active and archive log data sets known to DB2. DB2 uses this information to track the active and archive log data sets. DB2 also uses this information to locate log records to satisfy log read requests during normal DB2 system activity and during restart and recovery processing.

For any log, the list tells the RBA range (or LRSN range in a data sharing environment) in each data set. (The list specifies separate log data set name entries for each volume on which the log resides.) For the active log, the list also tells which are full and which are available for reuse. DB2 records data about the log data set in the BSDS each time a new archive log data set is defined or an active log data set is reused.

- A wrap-around inventory of all recent DB2 checkpoint activity. DB2 uses this information during restart processing.
- The distributed data facility (DDF) communication record. This record contains the DB2 location name, the virtual telecommunications access method (VTAM) LU name, and the password used to connect DB2 to VTAM. DB2 uses this information to establish the distributed database environment.
- The bootstrap dataset (BSDS) also includes a table of IP addresses. These uniquely identify a host within the TCP/IP network.

Because the BSDS is essential to recovery in the event of subsystem failure, during installation DB2 automatically creates two copies of the BSDS and, if space permits, places them on separate volumes.

For a more complete description of the functions and uses of the BSDS, see

Buffer Pools

Buffer pools, also known as *virtual buffer pools*, are areas of virtual storage used temporarily to store pages of table spaces or indexes. When an application program needs to access a row of a table, DB2 retrieves the page containing that row and places the page in a buffer. If the row is changed, the buffer must be written back to the table space. If the needed data is already in a buffer, the application program will not have to wait for it to be retrieved from DASD. The result is faster performance. The sizes of virtual buffer pools can be changed while DB2 is running. The result is greater flexibility. See Chapter 2 of *Command Reference* for details about the ALTER BUFFERPOOL command.

DB2 supports a second level of storage for each virtual buffer pool if your system meets the following requirements:

- A processor that supports MVPG hardware instruction and Asynchronous Data Mover Facility licensed internal code
- Expanded storage available in ES/9000 hardware
- MVS/ESA Version 4 Release 3 with the Special Programming Enhancement containing support for Asynchronous Data Mover Facility

The second level of storage, the *hiperpool*, is an extension to the virtual buffer pool. Virtual buffer pools hold the most frequently accessed data. Data in virtual buffer

pools that is not accessed frequently can be moved to its corresponding hiperpool—only one hiperpool can exist for each virtual buffer pool.

Hiperpools can span up to four *hiperspaces*, 2GB expanded storage areas. Using hiperspaces and hiperpools improves performance because you can cache up to 8GB to help avoid I/O operations.

When you install DB2, you make the following decisions about virtual buffer pools and hiperpools:

- Specify the number of 4KB virtual buffer pools and 32KB virtual buffer pools that you plan to use. You can have up to 50 4KB virtual buffer pools, and up to 10 32KB virtual buffer pools. The number of buffers within each pool is always less than or equal to the corresponding value specified on one of the buffer pool sizes panels (DSNTIP1, DSNTIP2).
- Specify whether you want a hiperpool to exist for a given virtual buffer pool. There can be only one hiperpool for each virtual buffer pool, and the sum of all hiperpools must not exceed 8GB of storage.

Another level of buffer pools is the group buffer pool, associated with the DB2 data sharing environment. For information relating to group buffer pools, see *Data Sharing: Planning and Administration*.

Communications Database

The *communications database* (CDB) is part of the catalog shipped with DB2. The distributed data facility (DDF) uses the CDB to map DB2 location names to VTAM LU names and TCP/IP IP addresses or domain names. The CDB also handles security translation requirements and communication requirements. The CDB is used to get information about communicating with other DB2 subsystems, or with remote locations that support Distributed Relational Database Architecture. Once your subsystem is installed, you must populate the CDB tables with unique information that enables your location to send and receive distributed data requests. For more information on the CDB, see .

Data Definition Control Support Database

The *data definition control support database* is automatically created during installation. This database is a user-maintained collection of tables used by data definition control support to restrict the submission of specific DB2 DDL statements to selected application identifiers (plans or collections of packages). Once this database is created, you must populate these tables to make use of this facility. The system name for this database is DSNRGFDB. For more information about DDL registration, see .

Resource Limit Facility Database

The *resource limit facility database* contains tables that limit the amount of processor time permitted for the execution of dynamic SELECT, UPDATE, DELETE, and INSERT SQL statements. You can establish a single limit for all users, different limits for individual users, or both. No limits apply to those with installation SYSADM or installation SYSOPR authority.

The limits are defined in one or more resource limit specification tables (RLST). One RLST is used for each invocation of the resource limit facility and is identified on the START RLIMIT command. If you are using DDF, the RLST contains columns so you can specify limits for primary authorization IDs, plan names, or

package names from other subsystems. The RLST also governs BIND authority. You can have more than one RLST, but only one RLST is active at any one time. The system name for this database is DSNRLST.

The RLST also contains a column that determines which mode of parallelism is disabled, if query CP parallelism or query I/O parallelism is possible. For more information on the RLST, see .

Work File Database

The *work file database* is used as storage for processing SQL statements that require working space. You can create a work file database using the CREATE DATABASE and CREATE TABLESPACE statements.

In a non-data sharing environment, the work file database is called DSNDB07. In a data sharing environment, each DB2 member in the data sharing group has its own work file database. One member of the data sharing group can have the name DSNDB07, but you can create a work file database with a more meaningful name. The default is DSN1. The recommended name is the DB2 subsystem name.

Control and Maintenance of DB2

DB2 is controlled and maintained by the following:

- Commands, which can be entered at a terminal or a MVS console
- Utility jobs, which run as standard MVS batch jobs.

Commands

DB2 is controlled by commands entered at a terminal or MVS console. The commands are divided into the following categories:

- DSN command and subcommands
- DB2 commands
- IMS commands
- CICS attachment facility commands
- MVS IRLM commands
- TSO CLIST commands.

For example, the command -START DB2 starts DB2. -STOP DB2 stops it. For a description of the steps used in operating DB2, and the commands used to implement them, see . For more information on commands, see Chapter 2 of *Command Reference*.

Utility Jobs

Many of the tasks of maintaining DB2 data, such as loading a table, copying a table space, or recovering a database to some previous point in time, are done by parts of DB2 called *utilities*.

The utilities run as batch jobs under MVS. *DB2 interactive* (DB2I) provides a simple way to prepare the job control language (JCL) for those jobs and to perform many other operations by entering values on panels. DB2I runs under TSO using ISPF services. To use DB2I, follow your local procedures for logging on to TSO, entering ISPF, and displaying the DB2I menu, shown in Figure 12 on page 1-38.

You control each operation by entering values that describe it on the panels provided. There are also help panels giving the syntax and examples of commands and utility control statements. To access the help panels, press the HELP PF key. (The HELP PF key can be set locally, but typically it is PF key 1.)

A utility control statement tells a particular utility what task to perform; many examples appear in this book. To run a utility job, first enter the control statement in a data set you use for input. Then invoke DB2I and select option 8, UTILITIES, on the DB2I Primary Option Menu. In some cases, you need other data sets; for example, the LOAD utility requires a data set containing the data that is to be loaded. For detailed instructions, see Section 1 of *Utility Guide and Reference* .

High Availability

It is not necessary to start or stop DB2 often. For nearly continuous operation, DB2 has been designed with the following capabilities:

- Online definition and modification of database and authorization descriptors.
- Online binding of application plans.
- Online changing of buffer pool and hiperpool sizes.
- Online execution of most utilities. For example:
 - You can recover online such structures as table spaces, partitions, data sets, a range of pages, a single page, and indexes.
 - You can recover several indexes or index partitions simultaneously to reduce recovery time.
 - You can read and update a table space while copying it.
 - You can reorganize table spaces and partitions separately. and read the data during the unload phase. You can specify the degree of access to your data during reorganization
- Availability of a table space (provided it is not explicitly stopped) after an I/O error, except for any portions that span the error ranges.
- Fewer pages are unavailable because of error conditions. For any structure that has a problem while applying a log record to the structure, DB2 adds the relevant pages and log ranges to the logical page list (LPL). This leaves only pages affected by the error condition unavailable.
- Improved reorganization time for a table with a low cluster ratio by specifying the SORTDATA parameter on the REORG utility.
- Using package versions permits binding while the applications continue to run. If there is a problem in the new application, the old program uses the old version of the package. When an application changes, only the programs that have changed need to be rebound.
- Continuing operation of DB2 after an I/O error writing a log record. On the active log, it moves to the next data set; on the archive log, it dynamically allocates another data set.
- Remote site disaster recovery methods that allow you to prepare for disasters that could cause a complete shutdown of your local DB2 system.

- Typical continuation of DB2 during restoration of dual operation of the bootstrap data set, active logs, and archive logs if degradation to single copy mode was necessary.
- DB2's data sharing function allows applications running on more than one DB2 subsystem to read and write to the same set of data concurrently. For details on the benefits of data sharing see *Data Sharing: Planning and Administration*.

To reduce the probability and duration of unplanned outages you should periodically back up and reorganize your data. Because these affect the availability of the databases, you should limit your use of, and understand the options of, utilities such as COPY and REORG.

#

The CONCURRENT and SHRLEVEL options of the COPY utility can minimize outages and improve availability of DB2 data during backup processing. If you use the SHRLEVEL CHANGE option, the data sets being copied are available during the entire copy operation. If you use SHRLEVEL REFERENCE, the availability of data sets depends on whether you also use the CONCURRENT option, which invokes DFSMS Concurrent Copy.

- If you specify SHRLEVEL REFERENCE but not CONCURRENT, data sets are unavailable until the copy operation is complete.
- If you specify SHRLEVEL REFERENCE and CONCURRENT, data is unavailable only until DFSMSdss finishes logical processing of the list of data sets to be copied.

For information on using DB2 utilities such as COPY and REORG, see *Utility Guide and Reference*.

Unplanned outages are difficult to avoid entirely. However, the time that elapses because of an unplanned outage can be minimized, and the occurrence of these outages can be reduced. When unplanned outages occur, you can use the RECOVER utility to restore a damaged DB2 structure. For more information on recovering data, see and Section 2 of *Utility Guide and Reference*.

It is important to avoid I/O errors on table spaces, indexes, logs, and the bootstrap data set because recovery of these errors (using the RECOVER utility) causes unplanned outages. For more information on I/O error recovery, see Section 2 of *Utility Guide and Reference*.

To ensure continuous availability, it is important to monitor the databases regularly. Monitoring measures the efficiency of your database, in both performance and space utilization. Most of your base tables and indexes are constantly being changed through updates, inserts, and deletions. Monitoring your space utilization can prevent problems. You can monitor and tune a database by using the RUNSTATS and STOSPACE utilities.

DB2 and the MVS Environment

DB2 operates as a formal subsystem of MVS/ESA. DB2 utilities run in the batch environment, and applications that access DB2 resources can run in the batch, TSO, IMS, or CICS environments. IBM provides attachment facilities to connect DB2 to each of these environments.

Address Spaces

DB2 requires several different address spaces for the following purposes:

- One for *database services*, DSN1DBM1, which manipulate most of the structures in user-created databases.
- One for *system services*, DSN1MSTR, which perform a variety of system-related functions.
- One for *distributed data facility*, DSN1DIST, which provides support for remote requests.
- One for the *internal resource lock manager (IRLM)*, IRLMPROC, which controls DB2 locking.
- One for *DB2-established stored procedures*, DSN1SPAS, which provides an isolated execution environment for user-written SQL programs at a DB2 server.
- Zero to many for *WLM-established stored procedures* to be handled in order of priority and isolated from other stored procedures running in other address spaces.
- At least one, possibly several, of the following types of *user address spaces*:
 - TSO
 - Batch
 - CICS
 - IMS dependent region
 - IMS control region

Figure 10 on page 1-35 shows how the address spaces relate to batch, TSO, IMS, and CICS user address spaces. Each user address space communicates with database services, system services, and distributed data facility address spaces. In addition, the IMS user address space communicates with the IRLM and IMS address spaces. The system services, database services, CICS, IRLM, IMS, stored procedures and distributed data facility address spaces communicate with each other as the arrows indicate.

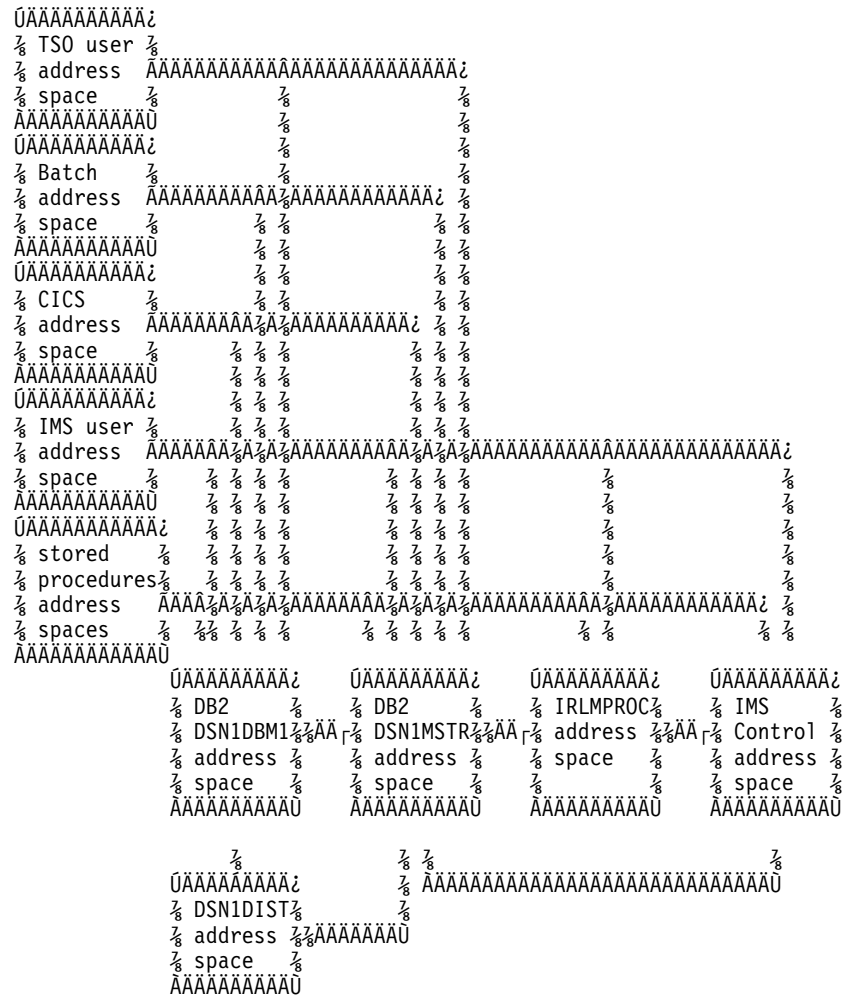


Figure 10. Relationship between DB2 Users and DB2-Related Address Spaces

DB2 and MVS

As a formal subsystem of MVS, DB2 uses:

- The MVS subsystem interface (SSI) protocols
- Key 7 operation and storage
- Synchronous cross-memory services for address space switching
- System Management Facilities (SMF) for statistics, accounting information, and performance data
- VTAM and TCP/IP for distributed data facility
- These reliability and serviceability features:
 - Functional recovery routines (FRR)
 - ESTAE recovery routines
 - SYS1.LOGREC
 - SYS1.DUMP

You can enter all DB2 commands from an authorized MVS console by using a subsystem command prefix (composed of 1 to 8 characters) at the beginning of the

command. The default subsystem command prefix is *-DSN1*. You can change the value when you install or migrate DB2.

DB2 and RACF

Resource Access Control Facility (RACF) can be used to control access to your MVS system. If you are using RACF Version 1 Release 9 or later, you can control access to the distributed data facility (DDF) based on the connecting partner's LU name. When users begin sessions with TSO, IMS, or CICS, their identities are checked to prevent unauthorized access to the system.

Recommendation: use RACF to run security checks on DB2 users and to protect DB2 resources. RACF provides effective protection for DB2 data by permitting only DB2-mediated access to DB2 data sets.

Much authorization to DB2 objects can be controlled directly from RACF. By using the access control authorization exit, a user can bypass some or most of DB2 authorization checking. For more information on writing exit routines, see Appendix B (Volume 2) of *Administration Guide*.

For a detailed description of security methods available in DB2,

DB2 and SMS

DFSMSdftp *storage management subsystem* (SMS) can be used to manage DB2 DASD data sets. Private data, image copies, and archive logs are possible candidates for space management with SMS. The decision to use SMS to manage DB2 data sets must be made with your site's storage administrator. For more information about using SMS to manage DB2 data sets, see *MVS Storage Management Library: Storage Management Subsystem Migration Planning Guide*.

Partitioned data set extended (PDSE), a feature of DFSMSdftp, provides a way to manage partitioned data sets. PDSE allows multiple access at a data set's member level rather than at the data set level, removing the concurrent access constraints of regular partitioned data sets. DB2 users who have MVS/ESA and DFSMSdftp installed should consider using PDSE data sets for their DBRM libraries.

PDSE data sets must be managed by SMS and stored on direct access storage devices. See *MVS/DFP: Managing Non-VSAM Data Sets* for information about differences of PDS and PDSE data sets and how to convert them.

DB2 and TSO Attachment Facility

The *Time Sharing Option* (TSO) attachment facility is required for binding application plans and packages and for executing several online functions that are provided with DB2.

Using the TSO attachment facility, you can access DB2 by running in either foreground or batch. You gain foreground access through a TSO terminal; you gain batch access by invoking the TSO terminal monitor program (TMP) from an MVS batch job.

Whether you access DB2 in foreground or batch, attaching through the TSO attachment facility and the DSN command processor makes access easier. The DSN command processor (DSN) executes as a TSO command processor. DB2 subcommands that execute under DSN are therefore subject to the command size


```

                                     DB2I PRIMARY OPTION MENU
====>_

Select one of the following DB2 functions and press ENTER.

 1 SPUFI                (Process SQL statements)
 2 DCLGEN                (Generate SQL and source language declarations)
 3 PROGRAM PREPARATION  (Prepare a DB2 application program to run)
 4 PRECOMPILE           (Invoke DB2 precompiler)
 5 BIND/REBIND/FREE     (BIND, REBIND, or FREE plans or packages)
 6 RUN                  (RUN an SQL program)
 7 DB2 COMMANDS         (Issue DB2 commands)
 8 UTILITIES            (Invoke DB2 utilities)
 D DB2I DEFAULTS       (Set global parameters)
 X EXIT                 (Leave DB2I)

PRESS:  END to exit          HELP for more information

```

Figure 12. DB2I Primary Option Menu

Because application programs are invoked under the TSO terminal monitor program, you can use ISPF to structure input and output. For information on using DB2 and ISPF together in an application, see Section 6 of *Application Programming and SQL Guide*.

Call Attachment Facility

Most TSO applications must use the TSO attachment facility, which invokes the DSN command processor. Together, DSN and TSO provide services such as automatic connection to DB2, attention key support, and translation of return codes into error messages. However, when using DSN services, your application must run under the control of DSN.

The *call attachment facility* (CAF) provides an alternative connection for TSO and batch applications needing tight control over the session environment. Applications using CAF can *explicitly* control the state of their connections to DB2 by using connection functions supplied by CAF. For more information on CAF, see Section 6 of *Application Programming and SQL Guide*.

DB2 and CICS

The *Customer Information Control System* (CICS) attachment facility provided with DB2 allows you to access DB2 from CICS. After you start DB2, you can operate DB2 from a CICS terminal. You can start and stop CICS and DB2 independently, and you can establish or terminate the connection between them at any time. You also have the option of allowing CICS to connect to DB2 automatically.

The CICS attachment facility also provides CICS applications with access to DB2 data while operating in the CICS environment. CICS applications, therefore, can access both DB2 data and CICS data. In case of system failure, CICS coordinates recovery of both DB2 and CICS data.

The CICS attachment facility uses standard CICS command-level services where needed; for example, EXEC CICS WAIT, EXEC CICS ABEND. A portion of the CICS attachment facility executes under the control of the transaction issuing the

attachment to the DB2 subsystem. A single CICS region can be connected to only one DB2 subsystem at a time.

System Administration and Operation with CICS

An authorized CICS terminal operator can issue DB2 commands to control and monitor both the attachment facility and DB2 itself. Authorized terminal operators can also start and stop DB2 databases.

Even though you perform DB2 functions through CICS, you need to have the TSO attachment facility and ISPF to take advantage of the online functions supplied with DB2 to install and customize your system. You also need the TSO attachment to bind application plans and packages.

There are significant changes to the CICS attachment facility with CICS Version 4. For more information on using CICS with DB2 .

DB2 and IMS

The *Information Management System* (IMS) attachment facility allows you to access DB2 from IMS. The IMS attachment facility receives and interprets requests for access to DB2 databases using exits provided by IMS subsystems. Usually, IMS connects to DB2 automatically with no operator intervention.

In addition to Data Language I (DL/I) and Fast Path calls, IMS applications can make calls to DB2 using embedded SQL statements. In case of system failure, IMS coordinates recovery of both DB2 and IMS data.

You can use DB2 with an extended recovery facility to facilitate recovery from an
IMS failure. To accomplish this, you must place all DB2 data sets on DASD shared
between the primary and alternate XRF systems. This enables DB2 to be manually
stopped on the primary system and started on the alternate system. You must then
ensure that the DB2 data sets on the shared DASD cannot be updated at the same
time by both the primary and alternate XRF systems. You might use a multi-system
DASD serialization function, such as Global Resource Serialization (GRS), or any
other means of shared DASD protection. For more information about XRF, and
IMS/ESA Administration Guide: System. For more information on global resource
serialization, see *MVS/ESA Planning: Global Resource Serialization*.

Figure 14 on page 1-41 shows the relationship between DB2 and IMS.

applications or periodically copy data back and forth between DB2 tables and IMS DBs.

See *Data Propagator NonRelational MVS/ESA Administration Guide* for more information about Data Propagator NonRelational.

System Administration and Operation with IMS

An authorized IMS terminal operator can issue DB2 commands to control and monitor DB2. The terminal operator can also start and stop DB2 databases.

Even though you perform DB2 functions through IMS, you need the TSO attachment facility and ISPF to take advantage of the online functions supplied with DB2 to install and customize your system. You also need the TSO attachment facility to bind application plans and packages.

DB2 and DL/I Batch

The DL/I batch support allows you to access both IMS data (DL/I) and DB2 data in the IMS batch environment. DL/I batch access allows:

- Access to DB2 and DL/I data from application programs.
- Coordinated recovery through a two-phase commit process.
- Use of the IMS extended restart (XRST) and symbolic checkpoint (CHKP) calls by application programs to coordinate recovery with IMS, DB2, and generalized sequential access method (GSAM) files.

For more information on DL/I batch, see Section 5 of *Application Programming and SQL Guide*.

DB2 and DDF

The *distributed data facility* (DDF) is an optional feature that allows a DB2 application to access data at other DB2s and at remote relational database systems that support IBM's Distributed Relational Database Architecture (DRDA). In addition, DDF allows applications running in a remote application requester environment that supports DRDA to access data in DB2 subsystems. Figure 15 on page 1-43 gives an overview of DDF support.

Substantial improvements in distributed database access have been achieved with the stored procedures function and distributed threads enhancements.

Stored procedures solve the problem of high processor and elapsed time costs that DRDA users experience during SQL processing when accessing data managed by DB2 for OS/390. This function introduces an SQL interface that allows an SQL requester to invoke user-written SQL programs, or *stored procedures*, at a DB2 server. Local DB2 applications or remote DRDA applications can issue the new SQL CALL statement to invoke a stored procedure. With a single send or receive operation, a series of SQL statements are invoked in the stored procedure, thus significantly decreasing the costs of distributed SQL statement processing.

Distributed threads enhancements allow you to have up to 25000 distributed threads connected to DB2 at the same time. This increase gives you room to grow your distributed applications; more applications can now connect to DB2 without delays.

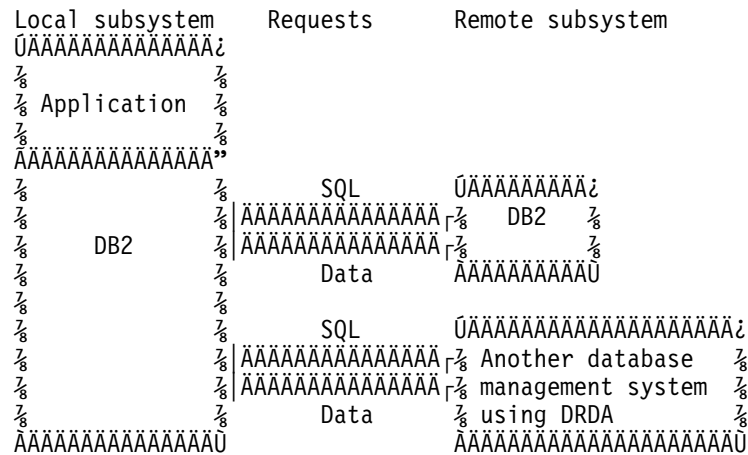


Figure 15. Overview of Distributed Database Support

The decision to access distributed data has implications for many DB2 activities: application programming, data recovery, authorization, and so on. For a discussion of these implications, see .

Data Sharing

DB2 takes advantage of the System/390 Parallel Sysplex, with its superior processing capabilities. By allowing two or more processors to share the same data, you can maximize performance while minimizing cost; improve system availability and concurrency; expand system capacity; and configure your system environment more flexibly. With data sharing, applications running on more than one DB2 subsystem can read from and write to the same set of data concurrently.

Figure 16. Data Sharing

Sharing DB2s must belong to a DB2 data sharing *group*. A data sharing group is a collection of one or more DB2 subsystems accessing shared DB2 data. Each DB2 subsystem belonging to a particular data sharing group is a *member* of that group. All members of a group use the same shared DB2 catalog and directory. DB2 data sharing is intended as a replacement for shared read-only data. Unlike read-only data sharers, all members of a data sharing group have equal and concurrent read and write access to databases.

Data sharing allows you to add another DB2 onto another central processor complex and access the same data through the new DB2. There is no need to manage copies or distribute data. All DB2s in the group have concurrent read and write access, and a single catalog and a single directory are used by all.

Data sharing allows you great flexibility in environment configuration. You can use separate MVS images, each tailored for its user set, sharing the same data. You can build your system incrementally as need dictates and set up members for use during peak times only.

For more information, see *Data Sharing: Planning and Administration*.

Section 2. Planning and Installing DB2

Chapter 2-1. Introduction to Installation and Migration	2-9
Features	2-9
Installation and Migration Steps Overview	2-10
SMP/E Steps Summary	2-11
Installation Steps Summary	2-11
Migration Steps Summary	2-12
Fallback Steps Summary	2-13
Remigration Steps Summary	2-14
Chapter 2-2. Estimating DB2 Storage Needs	2-15
DASD Storage for DB2 Subsystem	2-15
DASD Requirements for the DB2 Libraries and SMP/E Data Sets	2-16
DASD Requirements for the DB2 Catalog	2-18
DASD Requirements for the DB2 Directory	2-18
DASD Requirements for the Active Log Data Sets	2-18
DASD Requirements for the Bootstrap Data Sets	2-21
DASD Requirements for the Work File Database	2-21
DASD Requirements for the Default Database	2-23
DASD Requirements for the Dump Data Set Size	2-23
DASD Requirements for the System Databases	2-24
DASD Requirements for the Archive Log Data Sets	2-24
Using the Installation CLIST to Calculate Storage	2-24
DASD Storage for User Data	2-25
Calculating the Space Required for a Table	2-27
Calculating the Space Required for a Dictionary	2-28
Calculating the Space Required for an Index	2-29
Virtual Storage for Address Spaces	2-34
DB2 Distributed Data Facility Address Space (DSN1DIST)	2-35
IRLM Address Space (IRLMPROC)	2-35
DB2 System Services Address Space (DSN1MSTR)	2-35
DB2 Database Services Address Space (DSN1DBM1)	2-35
Allied Agent Address Space	2-36
Common Service Area	2-36
DB2-Established Stored Procedures Address Space (DSN1SPAS)	2-37
WLM-Established Stored Procedures Address Spaces	2-37
Virtual Storage for Storage Pools and Working Storage	2-38
Buffer Pool Size Calculation	2-38
Sort Pool Size Calculation	2-40
RID Pool Size Calculation	2-41
EDM Pool Size Calculation	2-41
Data Set Control Block Storage Size Calculation	2-45
Working Storage Calculation	2-46
Virtual Storage Below the 16MB Line	2-47
Real Storage	2-48
Chapter 2-3. Setting Up DB2 Online Help	2-51
Installing the Online Help Reader	2-52
Storage Requirements for the Online Help Reader	2-52
Copying Sample Installation Jobs for the Online Help Reader	2-54
Allocating the CSI Data Sets	2-56

Allocating the SMP/E Data Sets	2-56
Allocating the Online Reader Data Sets	2-56
Defining SMP/E DDDEF Entries	2-56
Performing the SMP/E RECEIVE Step for the Online Help Reader	2-56
Performing the SMP/E APPLY Step	2-56
Performing the Post-APPLY Link-Edit Step	2-57
Copying the Online Dictionary and Bookshelf List	2-57
Performing the SMP/E ACCEPT Step	2-57
Additional Installation Tasks	2-57
Unloading DB2 Books	2-59
Changing Book Data Set Names	2-61
Customizing BookManager READ/MVS	2-61
Verifying Online Help and Setting Exit Options	2-63
Using Online Help	2-65
Accessing Help with DB2 Online Help	2-65
Moving Around in the DB2 Online Help	2-65
Maintaining Online Help	2-67
Chapter 2-4. Loading DB2 Libraries	2-69
What IBM Sends You	2-69
What You Produce	2-70
SMP/E Step 1: Copy and Edit the SMP/E Jobs	2-72
Copying the SMP/E Jobs	2-73
Editing the SMP/E Jobs	2-73
SMP/E Step 2: Allocate Distribution and Target Libraries: DSNTIJAE	2-80
SMP/E Step 3: Run the RECEIVE Job: DSNTIJRC	2-81
SMP/E Step 4: Cleanup Job for Migration: DSNTIJUD (Optional)	2-82
SMP/E Step 5: Run the APPLY Job: DSNTIJAP	2-83
SMP/E Step 6: Run the ACCEPT Job: DSNTIJAC	2-84
SMP/E Step 7: Unload the Jobs for the DB2I Panels	2-84
SMP/E Step 8: Allocate Libraries for DB2I Panels	2-85
SMP/E Step 9: Run the RECEIVE Job for the DB2I Panels	2-85
SMP/E Step 10: Run the APPLY Job for the DB2I Panels	2-85
SMP/E Step 11: Run the ACCEPT Job for the DB2I Panels	2-86
SMP/E Step 12: Copy and Edit the SMP/E Jobs for DB2 REXX Language	
Support	2-86
SMP/E Step 13: Run the RECEIVE Job for DB2 REXX Language Support	2-86
SMP/E Step 14: Run the APPLY Job for DB2 REXX Language Support	2-86
SMP/E Step 15: Run the ACCEPT Job for DB2 REXX Language Support	2-86
SMP/E Step 16: Ensure Installation of Proper Maintenance	2-86
Finishing SMP/E Processing	2-87
Chapter 2-5. Installing, Migrating, and Updating System Parameters	2-89
Running the Installation CLIST	2-89
Making the DB2 ISPF Libraries Available to TSO	2-89
Invoking the CLIST	2-90
General Instructions	2-90
Directory of Panels	2-92
Directory of Panel Field Names	2-93
Online Book Data Set Names Panel: DSNTIPA0	2-98
Main Panel: DSNTIPA1	2-99
Recommended Approach for a New Installer	2-104
Data Parameters Panel: DSNTIPA2	2-105
Define Group or Member Panel: DSNTIPK	2-110

|

|

System Resource Data Set Names: DSNTIPH	2-112
Data Set Names Panel 1: DSNTIPT	2-116
Data Set Names Panel 2: DSNTIPU	2-121
Data Set Names Panel 3: DSNTIPQ	2-127
Data Set Names Panel 4: DSNTIPG	2-130
Data Set Names Panel 5: DSNTIPW	2-133
CICS Release Panel: DSNTIPV	2-135
Install DB2 CICS Datasets Panel: DSNTIP3	2-136
Sizes Panel: DSNTIPD	2-139
Thread Management Panel: DSNTIPE	2-145
Buffer pool Sizes Panel 1: DSNTIP1	2-150
Buffer pool Sizes Panel 2: DSNTIP2	2-152
Tracing Panel: DSNTIPN	2-154
Operator Functions Panel: DSNTIPO	2-158
Application Programming Defaults Panel 1: DSNTIPF	2-164
Application Programming Defaults Panel 2: DSNTIP4	2-170
IRLM Panel 1: DSNTIPI	2-174
IRLM Panel 2: DSNTIPJ	2-177
Protection Panel: DSNTIPP	2-181
MVS PARMLIB Updates Panel: DSNTIPM	2-186
Active Log Data Set Parameters: DSNTIPL	2-190
Archive Log Data Set Parameters: DSNTIPA	2-193
Databases and Spaces to Start Automatically Panel: DSNTIPS	2-199
Distributed Data Facility: DSNTIPR	2-201
Distributed Data Facility Panel: DSNTIP5	2-205
Stored Procedures Parameters Panel: DSNTIPX	2-207
Data Definition Control Support Panel: DSNTIPZ	2-209
Job Editing Panel: DSNTIPY	2-212
Install DB2—CLIST Calculations Panel 1: DSNTIPC	2-215
Install DB2—CLIST Calculations Panel 2: DSNTIPC1	2-219
Completing the CLIST Processing	2-219
Responding to Messages	2-219
Tailoring the Installation Jobs	2-220
Editing the Subsystem Parameters	2-223
Directory of Subsystem Parameters	2-223
The Update Process	2-228
Updating Parameters through the Update Selection Menu Panel:	
DSNTIPB	2-228
Updating Other Parameters	2-230
Chapter 2-6. Installing the DB2 Subsystem	2-233
Installation Step 1: Define DB2 to MVS: DSNTIJMV	2-233
DSNTIJMV Updates to SYS1.PARMLIB	2-234
DSNTIJMV Updates to SYS1.PROCLIB	2-237
Installation Step 2: Define the Integrated Catalog Facility Catalog and Alias:	
DSNTIJCA	2-238
Installation Step 3: Define System Data Sets: DSNTIJIN	2-239
Installation Step 4: Initialize System Data Sets: DSNTIJID	2-239
Installation Step 5: Define DB2 Initialization Parameters: DSNTIJUZ	2-240
Installation Step 6: Define User Authorization Exit Routines: DSNTIJEX	
(Optional)	2-241
Installation Step 7: Record DB2 Data to SMF (Optional)	2-242
Installation Step 8: Establish Subsystem Security (Optional)	2-243
Installation Step 9: Connect DB2 to TSO	2-243

	Making DB2 Load Modules Available to TSO and Batch Users	2-244
	Making DB2 CLISTs Available to TSO and Batch Users: DSNTIJVC	2-244
	Ensuring that PL/I Options are Available	2-245
	Making Panels, Messages, and Load Modules Available to ISPF and TSO	2-245
	Connecting DB2I Panels to the ISPF Main Panel	2-246
	Establishing DB2 Authorization IDs in TSO and RACF	2-250
	Installation Step 10: Connect IMS to DB2 (Optional)	2-250
	Installation Step 11: Connect CICS to DB2 (Optional)	2-250
	Installation Step 12: IPL MVS	2-251
	Installation Step 13: Start the DB2 Subsystem	2-251
	Installation Step 14: Define Temporary Work Files: DSNTIJTM	2-252
	Installation Step 15: Define and Bind DB2 Objects and User-Maintained Databases: DSNTIJSG	2-254
	Installation Step 16: Populate the User-Maintained Databases (Optional)	2-255
#	Installation Step 17: Bind the Packages for DB2 REXX Language Support: DSNTIJRX	2-256
#	Installation Step 18: Back Up the DB2 Directory and Catalog: DSNTIJIC	2-256
	Authorizing DB2 Users	2-257
	Installation Step 19: Verify the Installation	2-257
	Installing a Communications Network	2-257
	Installing a Second DB2 on the Same Operating System	2-257
	Planning Considerations	2-258
	Installation Procedure	2-259
	Considerations for Using Shared DASD	2-263
	Enabling stored procedures after installation	2-263
#	Enabling DB2-supplied stored procedures after installation	2-265
	Chapter 2-7. Migrating the DB2 Subsystem	2-267
	Migration Considerations	2-267
	WLM-established Stored Procedures Address Space	2-267
	GRANTs on SYSCAT.PROCEDURES Lost	2-268
	Selective Partition Locking	2-268
	Work File Database Size Calculations	2-268
	REPORT RECOVERY Changes	2-269
#	RUNSTATS Changes	2-269
	Entry SQL Conformance	2-269
	Catalog Column Changes	2-270
	Migrating a Member of a Data Sharing Group	2-270
	Release Coexistence	2-271
#	Migration Step 1: Run Link Checker on DB2 for MVS/ESA Version 4 Table Spaces (optional)	2-272
#	Migration Step 2: Determine Which Plans and Packages Are Invalid after Migration (Optional)	2-273
	Migration Step 3: Check for Consistency between Catalog Tables (Optional)	2-274
	Migration Step 4: Image Copy Directory and Catalog in Case of Fallback: DSNTIJIC	2-274
	Migration Step 5: Connect DB2 to TSO	2-275
	Making DB2 Load Modules Available to TSO and Batch Users	2-275
	Making DB2 CLISTs Available to TSO and Batch Users: DSNTIJVC	2-276
	Making Panels, Messages, and Load Modules Available to ISPF and TSO	2-276
	Migration Step 6: Connect IMS to DB2 (Optional)	2-278
	Migration Step 7: Connect CICS to DB2 (Optional)	2-278
	Migration Step 8: Stop DB2 for MVS/ESA Version 4 Activity	2-278

Migration Step 9: Back up Your DB2 for MVS/ESA Version 4 Volumes (Optional)	2-280
Migration Step 10: Define DB2 Initialization Parameter: DSNTIJUZ	2-280
DSNTIJUZ Actions	2-280
Additional Steps	2-280
Considerations for the BSDS	2-281
Migration Step 11: Establish Subsystem Security (Optional)	2-281
Migration Step 12: Define DB2 for OS/390 Version 5 to MVS: DSNTIJMV	2-281
DSNTIJMV Actions	2-282
Completing the Step	2-283
Migration Step 13: Define System Data Sets: DSNTIJIN	2-284
Migration Step 14: Define User Authorization Exit Routines: DSNTIJEX	2-284
Migration Step 15: IPL MVS	2-285
Migration Step 16: Start DB2 for OS/390 Version 5	2-285
Migration Step 17: Tailor DB2 for OS/390 Version 5 Catalog: DSNTIJTC	2-286
Migration Step 18: Ensure There Are No Problems with the Catalog (Optional)	2-287
Migration Step 19: Prepare Dynamic SQL Program: DSNTIJTM	2-287
Migration Step 20: Bind DB2 objects and User Maintained Database Activity: DSNTIJSG	2-287
Migration Step 21: Bind the Packages for DB2 REXX Language Support: DSNTIJRX	2-288
Migration Step 22: Image Copy DB2 for OS/390 Version 5 Catalog: DSNTIJIC	2-289
Migration Step 23: Verify Your DB2 for OS/390 Version 5 System	2-289
Running the DB2 for MVS/ESA Version 4 Sample Jobs (Optional)	2-289
Running the Version 5 Sample Jobs	2-290
Migration Step 24: Make Adjustments for Release Incompatibilities	2-290
Adjust Application Programs	2-290
Changes to DISPLAY DATABASE	2-292
SQL92 Entry Level Conformance	2-293
View WITH CHECK OPTION	2-293
SELECT Privilege for UPDATE and DELETE	2-293
REFERENCES Privilege on Columns	2-294
Migration Step 25: Convert Catalog and Directory Indexes (Optional)	2-294
Chapter 2-8. Falling Back and Remigrating	2-297
Falling Back	2-297
Fallback Considerations	2-297
Step 1: Run Phase 0 of the DB2 for OS/390 Version 5 Installation	
Verification Procedure	2-300
Step 2: Stop DB2 for OS/390 Version 5 Activity	2-300
Step 3: Reactivate DB2 for MVS/ESA Version 4 Code: DSNTIJFV	2-301
Step 4: Reconnect TSO, IMS, and CICS to DB2 for MVS/ESA Version 4	2-301
Step 5: Start DB2 for MVS/ESA Version 4	2-302
Step 6: Verify Fallback	2-303
Remigrating	2-303
Chapter 2-9. Verifying with the Sample Applications	2-305
Installation Verification Phases and Programs	2-306
Planning for Verification	2-308
Special Considerations for COBOL Programs	2-308
Special Considerations for C and C++ Programs	2-311
Special Considerations for PL/I Programs	2-312

Phase 0: Deleting the Sample Objects (DSNTEJ0)	2-312
Phase 1: Creating and Loading Sample Tables	2-313
Job DSNTEJ1	2-313
Job DSNTEJ1P	2-315
Phase 2: Testing the Batch Environment	2-316
Job DSNTEJ2A	2-316
Job DSNTEJ2C	2-317
Job DSNTEJ2D	2-317
Job DSNTEJ2E	2-318
Job DSNTEJ2F	2-318
Job DSNTEJ2P	2-319
Phase 3: Testing SPUFI, DRDA Access, Dynamic SQL, and TSO	2-319
Testing SPUFI	2-319
Running SPUFI at Remote non-DB2 Systems	2-320
Running Dynamic SQL and the ISPF/CAF Application	2-321
Jobs DSNTEJ3C and DSNTEJ3P:	2-321
Starting an Application in an ISPF/TSO Environment	2-323
Phase 4: Testing the IMS Environment	2-323
Jobs DSNTEJ4C and DSNTEJ4P	2-324
Starting an Application in an IMS Environment	2-326
Using the Phone Application in IMS	2-327
Phase 5: Testing the CICS Environment	2-327
Job DSNTEJ5A	2-327
Jobs DSNTEJ5C and DSNTEJ5P	2-328
Starting an Application in a CICS Environment	2-330
Using the Phone Application in CICS	2-331
Using CICS Storage-Handling Facilities	2-331
Phase 6: Accessing Data at a Remote Site	2-331
DRDA Access Sample	2-332
Job DSNTEJ6	2-333
DB2 Private Protocol Access Sample	2-333
Stored Procedure Samples	2-334
Stored Procedure Sample Without Result Set	2-335
Job DSNTEJ6S	2-335
Job DSNTEJ6P	2-335
Stored Procedure Sample With Result Set	2-336
Job DSNTEJ6T	2-336
Job DSNTEJ6D	2-337
Sample utilities stored procedure	2-338
Job DSNTEJ6U	2-339
Sample IMS ODBA stored procedure	2-339
Job DSNTEJ61	2-339
Job DSNTEJ62	2-340
Starting an Application in an ISPF/TSO Environment	2-340
The Sample Applications	2-341
Printing the Sample Application Listings	2-341
Specifying Values in the Sample Application Panels	2-341
Scenarios	2-344
The Project Application Scenario	2-345
The Organization Application Scenario	2-346
The Phone Application Scenario	2-351
The Distributed Organization Application Scenario	2-355
Edit Exit Routine	2-363
Huffman Compression Exit Routine	2-363

Sample Field Procedure	2-363
Dynamic SQL Statements (DSNTESA, DSNTESQ)	2-363
DSNTESA	2-364
DSNTESQ	2-365
Dynamic SQL Programs (DSNTIAD, DSNTPE2, DSNTIAUL)	2-366
DSNTIAD Sample Program	2-367
DSNTPE2 Sample Program	2-368
DSNTIAUL Sample Unload Program	2-368
Chapter 2-10. Connecting the IMS Attachment Facility	2-371
Making DB2 Load Modules Available to IMS	2-371
Defining DB2 to IMS	2-372
Defining New Programs and Transactions to IMS	2-375
Defining DB2 Plans for IMS Applications (Optional)	2-376
Generating a User Language Interface (Optional)	2-376
IMS Attachment Facility Macro (DSNMAPN)	2-377
Description	2-377
Chapter 2-11. Connecting the CICS Attachment Facility	2-379
Using the Attachment Facility for CICS Version 4 and Subsequent Releases	2-379
Using the Attachment Facility for CICS Version 3	2-380
Calculating Space Requirements for the CICS Attachment Facility	2-381
CICS Version 3	2-381
CICS Version 4	2-383
Estimating Storage Needed for CICS Attachment Threads	2-384
Defining the DB2 to CICS Connection Using the RCT	2-385
Updating the CICS System Tables	2-385
System Initialization Table Entries	2-386
Transaction Entries	2-387
Program Entries	2-390
PLT Processing	2-394
Providing CICS Support for DB2 Commands	2-395
Updating CICS Initialization JCL	2-396
Coordinating DB2 and CICS Security	2-396
Preparing CICS Applications for DB2	2-397
CICS Attachment Facility Macro (DSNCRCT)	2-398
DSNCRCT TYPE=INIT	2-399
DSNCRCT TYPE=COMD	2-406
DSNCRCT TYPE=POOL	2-407
DSNCRCT TYPE=ENTRY	2-409
DSNCRCT TYPE=FINAL	2-417
DSNCRCT TYPE=DSECT	2-417

Chapter 2-1. Introduction to Installation and Migration

This chapter introduces you to the features and steps needed to install or migrate to DB2 Version 5.

DB2 Version 5 comes with DB2 Installer—a DB2 installation tool that runs on your OS/2 workstation. DB2 Installer has a graphical interface and helps streamline some installation tasks. If you plan to use DB2 Installer to install or migrate, refer to the procedures in *DB2 Installer User's Guide*.

Installation is the process of preparing DB2 to operate as an MVS subsystem. *Migration* is the process of upgrading from a release of DB2 to a more current release. **You can only migrate to DB2 for OS/390 Version 5 from DB2 for MVS/ESA Version 4.** Whether you are installing or migrating, there are steps you must perform that are the same. Figure 17 on page 2-10 shows you the steps for a typical installation or migration. Each step refers you to a chapter in this book that explains that step in greater detail.

We recommend that you run your new DB2 subsystem in an OS/390 Release 3 environment to make the most of Version 5's features, although DB2 runs with MVS/ESA Version 4.3. You must migrate to MVS/ESA Version 4.3 before installing DB2 Version 5.

Before you begin installing or migrating, plan the amount of direct access storage and virtual storage you need. "Chapter 2-2. Estimating DB2 Storage Needs" on page 2-15 helps you with your decisions. Planning and coordinating with other DB2 subsystems is essential if you plan to install distributed data facility (DDF). For more information, see Section 2 (Volume 1) of *Administration Guide*. Review what values are needed for the parameters on the installation and migration panels. By planning in advance, filling in the parameters becomes an easier task. See "Running the Installation CLIST" on page 2-89 for help with your decisions.

Features

DB2 includes several features to help you perform the steps involved in installing or migrating to Version 5:

Installation and Migration Tools: DB2 provides a set of tools that automate the process of installing or migrating. These tools include:

- Most of the job control language (JCL) needed to install and migrate the product

This JCL constitutes the *installation and migration jobs*. Each of these jobs helps you perform a task when installing or migrating.

- The installation CLIST (command list) to help tailor the installation and migration jobs

This CLIST is also called the *migration CLIST*, or simply the *CLIST*. It contains the code necessary to tailor the jobs to your needs.

- A series of ISPF panels that you can use to pass information to the CLIST

With the Interactive Systems Productivity Facility (ISPF) and Interactive Systems Productivity Facility/Program Development Facility (ISPF/PDF), you

Figure 17. Installation and Migration Paths

can use a series of ISPF panels to pass parameter values to the CLIST. The CLIST uses these values to tailor the installation and migration jobs. This process is called the ISPF *tailoring session*.

- Sample applications to help determine if you installed or migrated DB2 correctly.

DB2 provides a set of sample programs and procedures that help you determine if DB2 is functioning correctly.

Minimal Assemblies: Because it is distributed as object code, DB2 requires few assemblies. You must perform an assembly to specify DB2 initialization parameters, but this requires only a few seconds.

Ability to Defer Decisions about DB2 Characteristics: DB2 allows you to specify many subsystem characteristics during DB2 operation. You can authorize users, define databases and tables, and tune DB2. Therefore, you can defer many decisions until after you finish installing or migrating DB2.

Ability to Update Installation and Migration Options: During the process of installing and migrating, DB2 uses ISPF panels to prompt you for many options. DB2 allows you to update most of these options without requiring you to reinstall or remigrate. You can accept the default for certain options and, after acquiring experience with DB2, tailor them to your needs.

Installation and Migration Steps Overview

Whether you are installing or migrating, you need to perform the following procedures:

- Install VTAM or TCP/IP network if using distributed data
- Set up a parallel Sysplex if using data sharing (see *System/390 MVS Sysplex Hardware and Software Migration* for information on setting up a sysplex)
- Load the DB2 libraries (do the SMP/E steps)

If you plan to use DB2's Call Level Interface (CLI), see *Call Level Interface Guide and Reference* for the additional installation jobs that you need to run.

If you plan to use DB2 for OS/390 Java™ Edition, see *Application Programming Guide and Reference for Java* for additional installation jobs that you need to run.

- Tailor the installation or migration jobs
- Install or migrate DB2
- Connect the DB2 attachment facilities
- Prepare DB2 for use
- Verify installation or migration

If you have problems during or after migration, you can perform the following procedures:

- Fall back to DB2 for MVS/ESA Version 4
- Remigrate to DB2 for OS/390 Version 5

After you have completed migration, we recommend that you avoid using new Version 5 facilities until you are certain that you will not need to fall back.

SMP/E Steps Summary

Before you begin installing or migrating DB2 you must unload the DB2 tapes or cartridges. Then, you edit and run SMP/E jobs. Following are the SMP/E steps you need to perform.

Before proceeding with these steps, refer to the *IBM DATABASE 2 Program Directory* shipped with DB2 for keyword specifications for Preventive Service Planning (PSP). Use Information/Access or the ServiceLink facility of IBMLink to check the most current information about DB2 and other products. Contact the IBM Support Center if you do not have access to IBMLink.

Table 2. Overview of SMP/E Steps

Step	Description	Job
1	Copy and edit the SMP/E jobs	IEBCOPY
2	Allocate the libraries	DSNTIJAE
3	Run the RECEIVE job	DSNTIJRC
4	Run the clean-up job (Optional)	DSNTIJUD
5	Run the APPLY job	DSNTIJAP
6	Run the ACCEPT job	DSNTIJAC
7	Unload the SMP/E jobs for the DB2I panels	IEBCOPY
8	Allocate the libraries for the DB2I panels	DSNTEJAE or DSNTNJAE
9	Run the RECEIVE job for the DB2I panels	DSNTEJRC or DSNTNJRC
10	Run the APPLY job for the DB2I panels	DSNTEJAP or DSNTNJAP
11	Run the ACCEPT job for the DB2I panels	DSNTEJAC or DSNTNJAC
12	Copy and edit the SMP/E jobs for DB2 REXX Language Support.	IEBCOPY
13	Run the RECEIVE job for DB2 REXX Language Support.	DSNTTJRC
14	Run the APPLY job for DB2 REXX Language Support.	DSNTTJAP
15	Run the ACCEPT job for DB2 REXX Language Support.	DSNTTJAC
16	Receive and apply any maintenance shipped with the product.	

Installation Steps Summary

After you have performed the SMP/E steps and followed the steps on page 2-90 to run the installation CLIST, you can edit and run the jobs that install your DB2 Version 5 subsystem. The following steps install DB2 Version 5.

Table 3 (Page 1 of 2). Overview of Steps for Installing DB2 Version 5

Step	Description	Job
1	Define DB2 Version 5 to MVS, and build cataloged procedures	DSNTIJMV

Table 3 (Page 2 of 2). Overview of Steps for Installing DB2 Version 5

Step	Description	Job
2	Optionally, define a new integrated catalog facility catalog and integrated catalog facility catalog alias	DSNTIJCA
3	Define DB2 data sets	DSNTIJIN
4	Initialize DB2 data sets	DSNTIJID
5	Define DB2 initialization parameters	DSNTIJUZ
6	Optionally, prepare authorization exit routines	DSNTIJEX
7	Optionally, set up for SMF recording	—
8	Optionally, establish subsystem security	—
9	Establish the DB2/TSO environment	DSNTIJVC
10	Optionally, connect IMS to DB2	—
11	Optionally, connect CICS to DB2	DSNTIJSU ¹
12	IPL MVS	—
13	Start DB2 Version 5	—
14	Define temporary table spaces, buffer pool, and hiperpool sizes	DSNTIJTM
15	Define and bind DB2 objects and user-maintained databases	DSNTIJSG
16	Optionally, populate the user-maintained databases and if you are using DDF populate the communications database (within the DB2 catalog)	—
17	Bind the packages for DB2 REXX Language Support, if you have installed DB2 REXX Language Support.	DSNTIJRX
18	Image copy DB2 catalog and DB2directory	DSNTIJIC
19	Run the installation verification procedure	DSNTEJxx

Note: ¹ Run this job for CICS 3.3 and earlier.

For a detailed description of the installation procedure, see “Chapter 2-6. Installing the DB2 Subsystem” on page 2-233.

Migration Steps Summary

After you have performed the SMP/E steps and followed the steps on page 2-90 to run the installation CLIST, you can edit and run the jobs that migrate your DB2 for MVS/ESA Version 4 subsystem to a DB2 for OS/390 Version 5 subsystem.

Migration to Version 5 includes the following steps:

Table 4 (Page 1 of 2). Overview of Steps for Migrating to DB2 Version 5

Step	Description	Job
1	Run DSN1COPY with the CHECK option on the catalog table spaces and invoke the link checker (DSN1CHKR) to check for broken links on your Version 4 subsystem.	—
2	Optionally, determine which plans and packages will be invalid after migration	—

Table 4 (Page 2 of 2). Overview of Steps for Migrating to DB2 Version 5

Step	Description	Job
3	Optionally, check for consistency between catalog tables	—
4	Image copy your Version 4 catalog	DSNTIJIC
5	Establish the DB2/TSO environment	DSNTIJVC
6	Optionally, connect IMS to DB2	—
7	Optionally, connect CICS to DB2	DSNTIJSU ¹
8	Stop DB2 Version 4	—
9	Back up Version 4 volumes	—
10	Define DB2 initialization parameters	DSNTIJUZ
11	Optionally, establish subsystem security	—
12	Define DB2 Version 5 to MVS and build cataloged procedures	DSNTIJMV
13	Define system data sets	DSNTIJIN
14	Optionally, prepare authorization exit routines	DSNTIJEX
15	IPL MVS	— ²
16	Start DB2 Version 5	—
17	Tailor the DB2 Version 5 catalog	DSNTIJTC
18	Optionally, invoke the link checker (DSN1CHKR) to check for broken links on Version 5	—
19	Prepare the dynamic SQL program and define buffer pool and hiperpool sizes	DSNTIJTM
20	Bind SPUFI and DCLGEN	DSNTIJSG
21	Bind the packages for DB2 REXX Language Support, if you have installed DB2 REXX Language Support.	DSNTIJRX
22	Image copy the Version 5 catalog	DSNTIJIC
23	Run Version 4 verification jobs Run Version 5 verification jobs	DSNTEJxx
24	Make adjustments for release incompatibilities	—
25	Convert Catalog and Directory Indexes (Optional)	DSNTIJXC

Note:

¹ Run this job for CICS 3.3.

² Optional if no PARMLIB updates exist or if early code is at the right level.

Fallback Steps Summary

Fallback is the process of returning to DB2 for MVS/ESA Version 4 after attempting or completing a migration to DB2 for OS/390 Version 5. You can fall back if a severe error occurs either during migration or during operation of DB2 Version 5.

This process applies only to those who are migrating to Version 5. It is described in “Chapter 2-8. Falling Back and Remigrating” on page 2-297. To fall back to Version 4, perform the following steps:

Table 5. Overview of Steps to Fall Back to DB2 for MVS/ESA Version 4

Step	Description	Job
1	Run Phase 0 of the Version 5 installation verification procedure (if possible)	DSNTEJ0
2	STOP DB2 Version 5	—
3	Rename the cataloged procedures	DSNTIJFV
4	Reconnect TSO, IMS, and CICS to Version 4	—
5	Start Version 4	—
6	Relink Version 4 DSNTIAR with applications	—
7	Run the Version 4 installation verification jobs	DSNTEJxx

Remigration Steps Summary

Migration to DB2 for OS/390 Version 5 after falling back to DB2 for MVS/ESA Version 4 (remigration) is simpler than migration. To remigrate to Version 5, perform the following steps:

Table 6. Overview of Steps for Remigration to DB2 Version 5

Step	Description	Job
1	Run DSN1COPY with the CHECK option on the catalog table spaces. Invoke the link checker (DSN1CHKR) to check for broken links on Version 4. Execute the queries in DSNTESQ to check for consistency between catalog tables.	DSN1CHKR
2	Image copy Version 4	DSNTIJIC
3	Stop Version 4	—
4	Reconnect TSO, IMS, and CICS to Version 5	—
5	Rename cataloged procedures	DSNTIJMV
6	Start DB2 Version 5	—
7	Optionally, image copy the Version 5 catalog	DSNTIJIC
8	Delete steps DSNTIJD, DSNTIJR, DSNTIJC, and DSNTIJG. In step DSNTIRU, delete all statements that are not needed to bind SPUFI and DCLGEN. Execute DSNTIJSG.	DSNTIJSG
9	Run Version 4 verification jobs. Run Version 5 verification jobs.	DSNTEJxx

In addition, you must perform some tasks manually. These tasks, as well as the tasks performed by the jobs listed above, are explained in “Remigrating” on page 2-303.

Chapter 2-2. Estimating DB2 Storage Needs

This chapter describes the following aspects of storage:

- “DASD Storage for DB2 Subsystem”
- “DASD Storage for User Data” on page 2-25
- “Virtual Storage for Address Spaces” on page 2-34
- “Virtual Storage for Storage Pools and Working Storage” on page 2-38
- “Real Storage” on page 2-48.

The parameters you specify when you run the installation CLIST affect the sizes of some data sets and the amount of virtual storage needed. All data sets are linear data sets with the exception of the bootstrap data set, which is a key-sequenced data set.

Data Facility Storage Management Subsystem (DFSMS) can be used to manage DB2 data sets. It provides automatic backup and recovery features, which might require DASD storage beyond what is estimated here. For more information, see *DFSMS/MVS: Storage Administration Reference for DFSMSdfp*.

This chapter explains how to calculate storage requirements for a small site, a medium site, a large site, and an extra large site. These models are based on the following assumptions:

- The **small** site supports a small number of DB2 users. The small site has about 100 plans, 50 application databases, and 500 tables.
- The **medium** site supports more extensive use of DB2 databases. The medium-sized site has about 200 plans, 200 application databases, and 2000 tables.
- The **large** site supports heavy use of DB2. The large site has about 400 plans, 400 application databases, and 4000 tables.
- The **extra large** site supports very heavy use of DB2. The extra large site has about 600 plans, 600 application databases, and 6000 tables.

When you first install DB2, we recommend that you follow one of these models. Later, you can modify parameters to better suit your needs.

Storage estimates for items specific to data sharing are in *Data Sharing: Planning and Administration* .

DASD Storage for DB2 Subsystem

This section assumes that, when running the installation CLIST, you accept the default values for the number of databases, tables, and application plans expected at your site. You specify these values on installation panel DSNTIPD.

If you do not accept the default values, you can calculate the storage needed for the DB2 data sets by using the information in “DASD Requirements for the Active Log Data Sets” on page 2-18. For other data sets, you can use the formulas in the CLIST. After calculating for each data set, you can calculate the total requirements.

To determine the storage requirements based on your DASD device model, check the values listed in Table 7 on page 2-16. The space requirements **do not** include space for user databases, image copies, archive logs, or temporary data sets that you create while installing or migrating.

Table 7. Estimated Space Requirements (in Cylinders) for DB2 by Site Size

Site Size	3380	3390	9340
Small	800	689	784
Medium	1217	1035	1160
Large	5178	4334	4757
Extra Large	9277	7750	8485

Table 8 provides storage requirements in megabytes (MB) for DB2 data sets. The calculations were done with more precision than is shown in the figure; this accounts for the discrepancy between individual amounts and totals. The estimated space requirements apply to 3380, 3390, and 9340 device types—there is no significant difference between the space requirements for the DB2 data sets on these device types. The DB2 libraries require a fixed amount of space, regardless of the size of your site. On the other hand, the DASD requirements for active logs and the DB2 catalog increase significantly as the size requirements for your site increase. You need additional space for archive logs, image copies, user databases, and other working data sets. Device-specific estimates in cylinders for items in Table 8 begin with Table 9 on page 2-17.

Table 8. Estimated Space Requirements (in Megabytes) for DB2 Data Sets by Site Size

Site Size	DB2 Libraries	DB2 Catalog	Directory	Active Logs	BSDS	Temp Database	Total
Small	316	61	27	20	1	24	449
Medium	316	180	72	40	1	24	633
Large	316	344	136	392	1	372	1561
Extra Large	316	509	200	784	1	580	2390

DASD Requirements for the DB2 Libraries and SMP/E Data Sets

Storage requirements for the DB2 libraries and SMP/E data sets are shown in Table 9 on page 2-17, Table 10 on page 2-17, and Table 11 on page 2-17.

Table 9. Estimated Space Requirements (in Cylinders) for DB2 Distribution Libraries

DB2 Distribution Libraries	3380	3390	9340
<i>prefix</i> .ADSNLOAD	81	71	81
<i>prefix</i> .ADSNMACS	76	67	76
<i>prefix</i> .ADSNENU ¹	5	5	5
<i>prefix</i> .ADSNDKF ¹	3	3	3
<i>prefix</i> .ADXRLoad	2	2	2
<i>prefix</i> .ADXRSAMP	1	1	1
Distribution Libraries Total¹	168	149	168

¹ The totals are actually dependent on whether you have selected to use both the English and the Kanji distribution libraries.

Table 10. Estimated Space Requirements (in Cylinders) for DB2 Target Libraries

Library	3380	3390	9340
DB2 Target Libraries			
<i>prefix</i> .SDSNCLST	5	4	5
<i>prefix</i> .SDSNEXIT	2	2	2
<i>prefix</i> .SDSNLINK	1	1	1
<i>prefix</i> .SDSNLOAD	49	43	49
<i>prefix</i> .SDSNMACS	15	13	15
<i>prefix</i> .SDSNSAMP	48	42	48
<i>prefix</i> .SDSNSPFM	1	1	1
<i>prefix</i> .SDSNSPFP	4	4	4
<i>prefix</i> .SDSNSPFT	1	1	1
<i>prefix</i> .SDSNPFPE ¹	5	5	5
<i>prefix</i> .SDSNPFPK ¹	3	3	3
<i>prefix</i> .ADSNENU ¹	1	1	1
<i>prefix</i> .ADSNDKF ¹	1	1	1
<i>prefix</i> .SDSNSPFS	1	1	1
<i>prefix</i> .SDSNDBRM	1	1	1
<i>prefix</i> .SDSNCHDR	1	1	1
<i>prefix</i> .SDXRRESL	1	1	1
<i>prefix</i> .SDXRSAMP	1	1	1
Target Libraries Total¹	141	126	141

¹ The actual totals depend on whether you have selected to use both the English and the Kanji target libraries.

These values are for initial allocation of the data sets. Some data sets grow with SMP activity, so your values could be much larger.

Table 11 (Page 1 of 2). Estimated Space Requirements (in Cylinders) for SMP/E Data Sets

SMP/E Data Sets	3380	3390	9340
SMPCSI	10	10	12

Table 11 (Page 2 of 2). Estimated Space Requirements (in Cylinders) for SMP/E Data Sets

SMP/E Data Sets	3380	3390	9340
SMPLOG	31	27	29
SMPLOGA	16	14	15
SMPMTS	1	1	1
SMPPTS	3	2	2
SMPSCDS	41	36	38
SMPSTS	3	2	2
SMPLTS	11	10	10
SMPTLIB	49	37	66
SMP/E Total	165	139	175

DASD Requirements for the DB2 Catalog

Storage requirements for the entire set of DB2 catalog data sets and their indexes are shown in Table 12.

Table 12. Estimated Space Requirements (in Cylinders) for the DB2 Catalog by Site Size

Site Size	3380	3390	9340
Small	89	74	81
Medium	265	221	241
Large	506	422	461
Extra Large	748	624	681

For information about how to change the size of catalog data sets after you install or migrate DB2, see Section 5 (Volume 2) of *Administration Guide*.

DASD Requirements for the DB2 Directory

Directory space depends mainly on the number of user databases, application plans and packages, and tables in the DB2 subsystem. Storage requirements for the DB2 directory are shown in Table 13.

Table 13. Estimated Space Requirements (in Cylinders) for the DB2 Directory by Site Size

Site Size	3380	3390	9340
Small	39	33	36
Medium	106	88	96
Large	200	167	182
Extra Large	294	245	267

DASD Requirements for the Active Log Data Sets

Active log data sets record significant events and data changes. They are periodically off-loaded to the archive log. So, the storage requirements for your active log data sets depend on how often DB2 data is changed at your site and how often DB2 off-loads those changes to the archive log.

If you change data frequently and off-load it to the archive log infrequently, you need a large amount of DASD space for the active log. If off-loading occurs once each day, under normal circumstances the active log data sets can hold the log records your subsystem produces during one day of processing.

These are the assumptions concerning each of the three models:

- The small site changes data 1800 times per hour, and the active log is off-loaded once each day.
- The medium-size site changes data 3600 times per hour, and the active log is off-loaded once each day.
- The large site changes data 36 000 times per hour, and the active log is off-loaded once each day.
- The extra large site changes data 72 000 times per hour, and the active log is off-loaded once each day.

As an example, here is how the DSNTINST CLIST calculates the amount of DASD space required by a small site:

1. During the ISPF tailoring session, you specified:
 - An archive period estimate of 24 hours—ARCHIVE LOG FREQUENCY parameter on installation panel DSNTIPL.
 - A data change rate estimate of 1800 changes per hour—UPDATE RATE parameter on installation panel DSNTIPL.
2. DB2 takes the size of a typical row as 200 bytes.
3. Other types of log records are comparatively small in size and are fixed in length. The length of each type depends on the information it contains.
4. The size of the active log, including DASD track overhead, is estimated as:

Data set size = (data change log record size)
× (data change rate per hour)
× (archive period)
= 400 bytes × 1800 per hour × 24 hours
+ data set allocation overhead
= 21MB
Dual logs = 42MB
Three data sets = 126MB

Data set allocation overhead is the difference between the allocated space and the data size requested in 4KB blocks. The change is caused by the difference between the space in 4KB blocks and the track size, which includes rounding up to a cylinder boundary. In this example, space is requested on a 3390, and 25 cylinders are allocated.

If you enabled data sharing, you generally need to have more DASD for the active log and archive the logs more frequently. See Chapter 6 of *Data Sharing: Planning and Administration* for more information.

If you accept the defaults of three active log data sets and dual logging, DB2 creates six active log data sets. For a typical production DB2 subsystem, you should have more than three active logs and dual logs. Other choices can lead to degraded performance when you encounter an IO error. You get better performance with larger active logs for other common problems, such as long

running updates. You can avoid some outages by having an adequate number of active logs for several hours of batch update processing.

Table 14 shows estimated storage requirements for active log data sets assuming dual logging. Table 15 shows the amount of space required for active log data sets on various IBM DASD devices. The estimates in both tables include DASD track overhead.

Table 14. Estimated Space Requirements (in Megabytes) for Active Log Data Sets by Site Size

Site Size	Archive Period (Hours)	Data Change Rate (Per Hour)	Space for Each Active Log Data Set (MB)	Total Space for Six Active Log Data Sets (MB)
Small	24	1800	20	120
Medium	24	3600	40	240
Large	24	36 000	392	2352
Extra Large	24	72 000	784	4704

Table 15. Estimated Space Requirements (in Cylinders) for the Active Log by Site Size

Site Size	3380	3390	9340
Small	174	150	162
Medium	348	294	318
Large	3462	2886	3144
Extra Large	6918	5766	6288

Some other considerations for the size of your active log data sets include:

- Tape utilization

When you archive to the media type listed in the following table, the numbers listed under “Estimated Planning Size” are suggested sizes for your active logs. If the size of an active log data set is small compared to the size of a tape, tape utilization is fairly low.

Table 16. Estimated Active Log Planning Size

Log Media	Estimated Planning Size
6250 BPI tape	100MB
3590 High-Performance Tape Subsystem	10GB
3480 cartridge	200MB or more
4mm cartridge (60m)	1.3GB
4mm cartridge (90m)	2.0GB
4mm cartridge (120m)	4.0GB

Using larger block sizes for archive logs can increase these figures by up to 40%. You specify block size for archive logs with the BLOCK SIZE field on installation panel DSNTIPA. Compression on the newer cartridge units can also increase these figures substantially. Using compression on the tape units will

encourage you to have larger active logs, controls for long-running updates, and to archive to disk with DFSMSHsm migration to tape.

- Checkpoint frequency

The CHECKPOINT FREQ field on panel DSNTIPN specifies the number of consecutive log records written between DB2 system checkpoints. It balances between the overhead needed for frequent subsystem checkpoints and the time to restart a DB2 subsystem after a termination without a quiesce. If the checkpoint value is more than 300 000, then the time needed to restart DB2 after a termination without a quiesce can grow to over 30 minutes. The recommended values for the checkpoint frequency are in the range of 25 000 to 500 000.

- Number of tables defined for data capture

When tables are defined with DATA CAPTURE CHANGES, the entire before-image of an updated row is captured on the log. This can represent an increase in log data compared to tables that are not defined with DATA CAPTURE CHANGES, depending on whether the table contains fixed or variable length rows. For information on what is logged for updated rows in both data-capture and non-data-capture tables, see Appendix C (Volume 2) of *Administration Guide*.

DASD Requirements for the Bootstrap Data Sets

Each bootstrap data set (BSDS) requires 0.5MB for each data set. If you are installing, DB2 automatically allocates two copies of the BSDS. If you are migrating, Version 5 adopts the BSDS characteristics you specified for DB2 for MVS/ESA Version 4. That is, if you specified two copies of the BSDS data sets for Version 4, you will have two copies for Version 5. Keeping two copies of the BSDS data set is strongly recommended. The total space requirement is about 1MB for both BSDSs. The BSDSs at any size site require about 2 cylinders, regardless of device type.

DASD Requirements for the Work File Database

The work file database is used as storage for processing SQL statements that require working storage. Table 17 shows the DASD requirement estimates for the temporary work files in the work file database. For additional migration considerations when running DSNTIJTC please refer to “Chapter 2-7. Migrating the DB2 Subsystem” on page 2-267. This chapter deals with specific migration considerations that you need for calculating the size of your work file database.

Table 17. Estimated Space Requirements (in Cylinders) for the Work File Database by Site Size

Site Size	3380	3390	9340
Small	35	29	32
Medium	35	29	32
Large	547	456	497
Extra Large	854	712	776

You might need more storage for the work file database if you have a great amount of data to sort and a great amount of concurrent sort activity. If you are sorting compressed data, allow for the same amount of storage that you would need if the data were not compressed. The maximum amount of storage you would need is

enough to satisfy the requirements of the largest combination of concurrent activities that use the work file database. The amount of storage required for a sort depends on the following variables:

- Data size
- Sort key size

You can estimate the total amount of work file space needed to perform the sort as follows:

- Let MIN be the operation of selecting the lowest value from a set of values.
- Let FLOOR be the operation of discarding the decimal portion of a real number.
- Let CEILING be the operation of rounding a real number up to the next highest integer.
- Let *data* be the total data length in bytes.
- Let *key* be the total length of the sort key.
- Let *prefix* be the 6 byte header.
- Let *rows* be the total number of rows being sorted.

Then calculate as follows:

$$\begin{aligned} \text{Records per page} &= \text{MIN}(\text{MAXROWS}, \text{FLOOR}(4076 / (\text{data} + \text{key} + \text{prefix}))), \\ &\text{but cannot exceed 255 (the value of MAXROWS)} \\ \text{Total pages} &= \text{CEILING}(\text{rows} / \text{Records per page}) \\ \text{Total segments} &= \text{CEILING}(\text{Total pages} / 24) \end{aligned}$$

This tells you how much storage is needed in the work file database after sort processing. However, if a merge phase was required during sort processing, an additional intermediate copy of the records might exist at any given time. For most subsystems, it is safe to assume that about half of the records involved in a sort have two copies. Therefore, a multiplier value of 1.5 is safe. If you want to be conservative, choose 2 for your multiplier value. Therefore, the amount of storage used in the work file database during sort processing can vary from 1 to 2 times the storage needed after sort processing. The actual storage used might also increase if you have little buffer pool storage available.

To determine the number of tracks needed, convert the number of pages into bytes and divide the result by the number of bytes per unit. Let *r* be the number of 4096 byte records per track and *safety_factor* anywhere from 1.5 to 2.0. For 3390 DASD, *r* is 12. For 3380 and 9340, *r* is 10.

$$\text{Tracks} = \text{CEILING}(\text{Total pages} / r) \times \text{safety_factor}$$

Example 1: Consider a table (TABLE1) containing 45327 rows, for which you want to create a **nonunique type 1 index** on COL1 CHAR(3) NOT NULL, COL2 CHAR(4), COL3 VARCHAR(20), and COL4 SMALLINT. Determine the amount of temporary storage needed to create this index as follows:

- $\text{Data} = 3 + (4 + 1) + (20 + 1) + (2 + 1) + 4 = 36$
- $\text{Key} = 36$ (data plus RID is key for CREATE INDEX)
- $\text{Rows} = 45327$
- $\text{Records per page} = \text{MIN}(\text{MAXROWS}, \text{FLOOR}(4076 / (36 + 36 + 6))) = 52$
- $\text{Total pages} = \text{CEILING}(45327 / 52) = 872$
- $\text{Segments} = \text{CEILING}(872 / 24) = 37$

- $Tracks = \text{CEILING}(872 / 12) \times 1.5 = 111$

Example 1 is a data page calculation for storing index keys in the work file database. For this example, 111 tracks of a 3390 storage device are needed. The 2-byte length field of a VARCHAR column is not a part of *data* for CREATE INDEX, the RID field is a part of *data*, and the key includes the entire data portion, including the RID.

Example 2: Consider TABLE1 again and the following SQL query:

```
SELECT COL1,COL2,COL3,COL4
   FROM TABLE1
   ORDER BY COL2,COL3,COL1;
```

This query requires a sort. Determine the amount of temporary storage required for this table as follows:

- $Data = 3 + (4 + 1) + (20 + 2 + 1) + (2 + 1) = 34$
- $Key = (4 + 1) + (20 + 1) + 3 = 29$
- $Rows = 45327$
- $Records\ per\ page = \text{MIN}(\text{MAXROWS}, \text{FLOOR}(4076 / (34 + 29 + 6))) = 59$
- $Total\ pages\ (final\ result) = \text{CEILING}(45327 / 59) = 769$
- $Segments\ (final\ result) = \text{CEILING}(769 / 24) = 33$
- $Total\ pages\ (during\ processing) = \text{CEILING}(1.5 \times 769) = 1154$
- $Segments\ (during\ processing) = \text{CEILING}(1.5 \times 35) = 53$
- $Tracks = \text{CEILING}(1238 / 12) = 104$

For this example which is a table calculation, 104 tracks of a 3390 storage device are needed. The 2 byte length field of a VARCHAR column is a part of *data* for CREATE INDEX, the RID field is not a part of *data*, and the key does not include the entire data portion.

The sort summary trace record, IFCID 0096, can be used to simplify some of the calculations. This record shows the number of records sorted, the sort record size (*data + key*), and whether or not a merge phase was required for an individual sort request. For information on the trace facility of DB2, see Section 5 (Volume 2) of *Administration Guide* .

DASD Requirements for the Default Database

The size of the default database depends on column lengths, page sizes, and index column lengths. The estimated size of your data, multiplied by 2, usually provides an adequate planning estimate for default database size.

DASD Requirements for the Dump Data Set Size

We recommend these guidelines for the dump data sets:

- At least 2 dump data sets
- Approximately 200 cylinders of 3390 DASD space for each SYS1.DUMPxx data set defined.
- 3.25MB of DB2 volatile summary storage data

This summary data is usually enough to diagnose most problems. In addition to summary data, DB2 also requests MVS SDUMP to provide these additional storage areas if enough space is available in the dump data set:

- DB2 system services address space

- DB2 database services address space
- DB2 distributed data facility address space
- Allied address space of the failing allied task
- IRLM address space when in a data sharing environment

DB2 passes these parameters to the MVS SDUMP service aid through the SDATA keyword: SQA, ALLPSA, LSQA, SUMDUMP, and CSA (subpools 231 and 241). Refer to *MVS/ESA Programming: Assembler Services Reference* for more information on the MVS SDUMP service aid.

After DB2 SVC dump processing is complete, MVS message IEA911E indicates whether enough space was available in the dump data set to contain the requested storage areas. If this message indicates that a partial dump was taken, but the 3.25MB of summary storage is available in the dump, this dump is probably enough for problem diagnosis. Otherwise, IBM might request that you re-create the problem if storage areas required for problem determination are not included in the dump.

DASD Requirements for the System Databases

If you are installing or migrating, DB2 automatically creates the resource limit facility database and the DB2 Connect (formerly data definition control (DDCS)) database. The storage requirements for these databases depend completely on the amount of user data.

DASD Requirements for the Archive Log Data Sets

If you decide to place the archive log data sets on DASD, you need to reserve enough space on those devices. The active log data set and the BSDS are both written to the same location. Therefore, you must reserve enough storage for the active log and the BSDS. Use the information found in “DASD Requirements for the Active Log Data Sets” on page 2-18 and “DASD Requirements for the Bootstrap Data Sets” on page 2-21 to determine these sizes, or see the messages generated by the CLIST on page 2-215. The amount of storage required for the logs and BSDSs combined is found in the messages for Volume Serial 5 (DSNV05) and Volume Serial 6 (DSNV06).

The installation CLIST uses the amount of space computed for the active log data sets for archive primary and secondary space. This size is computed by taking the active log data sets in bytes and dividing this number by the block size specified on installation panel DSNTIPA. Primary space for the archive log is the same as for the active log. Secondary space is small and is used if it is needed for cylinder rounding differences on different devices.

Attention:

Do not specify DFSMS compression for archive logs on DASD. If you do, you receive log read failures when attempting to recover from the archive logs.

Using the Installation CLIST to Calculate Storage

If you choose not to use the estimates for the model sites, you can use the detailed information for DASD storage estimates in the installation CLIST. We recommend you use the model site estimates the first time you install DB2. Use the model approach described in “DASD Storage for DB2 Subsystem” on page 2-15 to estimate DB2 DASD use. After your site has some experience in operating DB2, you can recalculate your DASD estimates.

The CLIST contains the algorithms that DB2 uses to calculate storage based on the parameters that you supply during installation or migration. You can use these algorithms to calculate the storage needs of your site on a data-set-by-data-set basis.

To see the algorithms that DB2 uses, print or edit the CLIST. You can run the CLIST to calculate the sizes. Installation panel DSNTIPC on page 2-215 displays the storage sizes calculated by the CLIST.

DASD Storage for User Data

You can use DB2 Estimator to calculate space estimates for tables, indexes, and factors listed below.

The amount of DASD space you need for your data is not just the number of bytes of data; the true number is some multiple of that. That is,

$$\text{Space required} = M \times (\text{bytes of data})$$

The multiplier M depends on your circumstances. It includes factors that are common to all data sets on DASD, as well as others that are peculiar to DB2. It can vary significantly, from a low of about 1.25, to 4.0 or more. For a first approximation, set M=2, and skip to “Calculating the Space Required for a Table” on page 2-27.

For more accuracy, calculate M as the product of the following factors:

Factor	Allows for
Record overhead	Eight bytes of record header and control data, plus space wasted for records that do not fit exactly into a DB2 page. For the second consideration, see Section 2 (Volume 1) of <i>Administration Guide</i> . The factor can range from about 1.01 (for a careful space-saving design) to as great as 4.0. A typical value is about 1.10.
Free space	Space intentionally left empty to allow for inserts and updates. You can specify this factor on the CREATE TABLESPACE statement; see Section 2 (Volume 1) of <i>Administration Guide</i> for more information. The factor can range from 1.0 (for no free space) to 200 (99% of each page used left free, and a free page following each used page). With default values, the factor is about 1.05.
Unusable space	Track lengths in excess of the nearest multiple of page lengths. DB2 uses 4KB pages, which are blocked to fit as many pages as possible on a track. The table below shows the track size, number of pages per track, and the value of the unusable-space factor for several different device types.

Table 18. Unusable Space Factor by Device Type

Device Type	3380	3390	9340
Track size	47476	56664	46456
Pages per track	10	12	10
Factor value	1.16	1.15	1.03

Data set excess	Unused space within allocated data sets, occurring as unused tracks or part of a track at the end of any data set. The amount of unused space depends upon the volatility of the data, the amount of space management done, and the size of the data set. Generally, large data sets can be managed more closely, and those that do not change in size are easier to manage. The factor can range without limit above 1.02. A typical value is 1.10.
Indexes	Storage for indexes to data. For data with no indexes, the factor is 1.0. For a single index on a short column, the factor is 1.01. If every column is indexed, the factor can be greater than 2.0. A typical value is 1.20. For further discussion of the factor, see “Calculating the Space Required for an Index” on page 2-29.

Table 19 shows calculations of the multiplier M for three different database designs:

- The “tight” design is carefully chosen to save space and allows only one index on a single, short field.
- The “loose” design allows a large value for every factor, but still well short of the maximum. Free space adds 30% to the estimate, and indexes add 40%.
- The “medium” design has values between the other two. You might want to use these values in an early stage of database design.

In each design, the device type is assumed to be a 3390. Therefore, the unusable-space factor is 1.15. M is always the product of the five factors.

Table 19. Calculations for Three Different Database Designs

Factor	“Tight” Design	“Medium” Design	“Loose” Design
Record overhead ×	1.02	1.10	1.30
Free space ×	1.00	1.05	1.30
Unusable space ×	1.15	1.15	1.15
Data set excess ×	1.02	1.10	1.30
Indexes =	1.02	1.20	1.40
Multiplier M	1.22	1.75	3.54

In addition to the space for your data, external storage devices are required for:

- Image copies of data sets, which can be on tape
- System libraries, system databases, and the system log
- Temporary work files for utility and sort jobs

A rough estimate of the additional external storage needed is three times the amount calculated above (space for your data) for DASD storage.

Calculating the Space Required for a Table

This section helps you calculate the space required for a table. Space allocation parameters are specified in kilobytes.

You can reduce the space required for a table by using data compression if your system meets the requirements. To find out how much space you can save by compressing your data, run the DSN1COMP utility on your DB2 for MVS/ESA Version 4 data sets. Message DSN1940I of DSN1COMP reports an estimate of the percentage of kilobytes that would be saved by using data compression. See Section 3 of *Utility Guide and Reference* for more information on the DSN1COMP utility.

The DASD saved by data compression is countered by the DASD required for a *dictionary*. Every compressed table space or partition requires a dictionary—see “Calculating the Space Required for a Dictionary” on page 2-28 to figure the DASD requirements and the virtual storage requirements for a dictionary.

For a table to be loaded by the LOAD utility, the value can be estimated as follows:

- Let FLOOR be the operation of discarding the decimal portion of a real number.
- Let CEILING be the operation of rounding a real number up to the next highest integer.
- Let *number of records* be the total number of records to be loaded.
- Let *average record size* be the sum of the lengths of the fields in each record, using an average value for varying-length fields, and including the following amounts for overhead:

- 8 bytes for the total record
- 1 byte for each field that allows nulls
- 2 bytes for each varying-length field

If the average record size is less than 32, use 32. See the CREATE TABLE statement in Chapter 6 of *SQL Reference* for information on how many bytes are required for different column types.

- Let *percsave* be the percentage of kilobytes saved by compression (as reported by the DSN1COMP utility in message DSN1940I)
- Let *compression ratio* be *percsave*/100

Then calculate as follows:

1. *Usable page size* is the page size less 22 bytes of overhead (that is, 4074 for 4KB pages or 32746 for 32KB pages) multiplied by $(100-p) / 100$, where *p* is the value of PCTFREE. If your average record size is less than 16, then usable page size is 255 (maximum records per page) multiplied by average record size multiplied by $(100-p) / 100$.
2. *Records per page* is $\text{MIN}(\text{MAXROWS}, \text{FLOOR}(\text{usable page size} / \text{average record size}))$, but cannot exceed 255 and cannot exceed the value you specify for MAXROWS.
3. *Pages used* is $2 + \text{CEILING}(\text{number of records} / \text{records per page})$.
4. *Total pages* is $\text{FLOOR}(\text{pages used} \times (1 + fp) / fp)$, where *fp* is the (nonzero) value of FREEPAGE. If FREEPAGE is 0, then *total pages* is equal to *pages used*. (See “free space” on page 2-25 for more about FREEPAGE.) If you are

using data compression, you need additional pages to store the dictionary. See “Calculating the Space Required for a Dictionary” on page 2-28 to figure how many pages the dictionary requires.

5. Estimated number of kilobytes required for a table:

- **If you do not use data compression**, the estimated number of kilobytes is $total\ pages \times page\ size\ (4KB\ or\ 32KB)$.
- **If you use data compression**, the estimated number of kilobytes is $(total\ pages \times page\ size\ (4KB\ or\ 32KB) \times (1 - compression\ ratio))$.

For example, consider a table space containing a single table with the following characteristics:

Number of records = 100000
Average record size = 80 bytes
Page size = 4KB
PCTFREE = 5 (5% of space is left free on each page)
FREEPAGE = 20 (one page is left free for each 20 pages used)
MAXROWS = 255

If the data is not compressed, you get the following results:

Usable page size = $4074 \times 0.95 = 3870$ bytes
Records per page = $\text{MIN}(\text{MAXROWS}, \text{FLOOR}(3870 / 80)) = 48$
Pages used = $2 + \text{CEILING}(100000 / 48) = 2085$
Total pages = $\text{FLOOR}(2085 \times 21 / 20) = 2189$
Estimated number of kilobytes = $2189 \times 4 = 8756$

If the data is compressed, multiply the estimated number of kilobytes for an uncompressed table by $(1 - compression\ ratio)$ for the estimated number of kilobytes required for the compressed table.

Calculating the Space Required for a Dictionary

This section helps you calculate the DASD space required by a *dictionary* and the virtual storage required in the DSN1DBM1 address space when a dictionary is read into storage from a buffer pool. A dictionary contains the information used for compressing and decompressing the data in a table space or partition, and it resides in that table space or partition. You can skip this section if you are not going to compress data. Space allocation parameters are specified in pages (either 4KB or 32KB).

DASD requirements

This section helps you calculate the DASD requirements for a dictionary associated with a compressed nonsegmented table space and for a dictionary associated with a compressed segmented table space.

For a nonsegmented table space, the dictionary contains 4096 entries in most cases. This means you need to allocate an additional sixteen 4KB pages or two 32KB pages. Although it is possible that your dictionary can contain fewer entries, we recommend allocating enough space to accommodate a dictionary with 4096 entries. For 32KB pages, 1 segment (minimum of 4 pages) is sufficient to contain the dictionary. Refer to Table 20 on page 2-29 to see how many 4KB pages or 32KB pages to allocate for the dictionary of a compressed nonsegmented table space.

<i>Table 20. Pages Required for the Dictionary of a Compressed Non-Segmented Table Space</i>				
Table Space Page Size (KB)	Dictionary Size (number of entries)			
	512	1024	2048	4096
4	2	4	8	16
32	1	1	1	2

For a segmented table space, the size of the dictionary depends on the size of your segments. Again, it is best to assume 4096 entries. Use Table 21 to see how many 4KB pages to allocate for the dictionary of a compressed segmented table space.

<i>Table 21. Pages Required for the Dictionary of a Compressed Segmented Table Space</i>				
Segment Size (4KB pages)	Dictionary Size (number of entries)			
	512	1024	2048	4096
4	4	4	8	16
8	8	8	8	16
12	12	12	12	24
16	16	16	16	16
20	20	20	20	20
24	24	24	24	24
28	28	28	28	28
≥32	Segment size	Segment size	Segment size	Segment size

Virtual storage requirements

You can calculate how much storage is needed in the DSN1DBM1 address space for each dictionary with this formula:

$$\text{dictionary size (number of entries)} \times 16 \text{ bytes}$$

When a dictionary is read into storage from a buffer pool, the *whole* dictionary is read, and it remains there as long as the compressed table space is being accessed.

Calculating the Space Required for an Index

For a more accurate estimate of required space use DB2 Estimator. The simplified
 # formulas in this book provide a rough estimate of space needed. Space allocation
 parameters are specified in kilobytes. For an index to be loaded by the LOAD
 utility, the value can be estimated as shown below.

| The storage required for an index, newly built by the LOAD utility, depends on the
 | type of index and number of index pages at all levels. That, in turn, depends on
 whether the index is unique. The numbers of leaf pages (index pages that point
 directly to the data in your tables) and of nonleaf pages (index pages that contain
 the page number and the highest key of each page in the next-level index) are
 calculated separately.

These simplified index calculations are intended only to assist you estimate the storage required for an index. Because there is no way to predict the exact number of duplicate keys that can occur in an index, the results of these calculations are not absolute. It is possible, for example, that for a nonunique index, more index entries than the calculations indicate might fit on an index page.

The following calculations are divided into these cases:

- Type 1 Indexes
 - A unique index with one subpage per page
 - A unique index with more than one subpage per page
 - A nonunique index with one subpage per page
 - A nonunique index with more than one subpage per page
- Type 2 Indexes
 - A unique index
 - A nonunique index

In the following calculations, let:

- k = the length of the index key. The length of the index key is the sum of the lengths of all the columns of the key, plus the number of columns that allow nulls.
- n = the average number of data records per distinct key value of a nonunique index. For example:
 - a = number of data records per index
 - b = number of distinct key values per index
 - s = the proportion of available space (equal to $(100-p)/100$, where p is the value of PCTFREE)
 - $n = a / b$
- f = the value of PCTFREE.
- p = the value of FREEPAGE.
- # r = record identifier (RID) length. Let $r = 4$ for indexes on non-LARGE table spaces and $r = 5$ for indexes on LARGE table spaces (type 2 only).
- # FLOOR = the operation of discarding the decimal portion of a real number.
- CEILING = the operation of rounding a real number up to the next highest integer.
- MAX = the operation of selecting the highest integer value.

Type 1 Index Calculations

If you specified a FREEPAGE parameter (f) greater than zero on the CREATE INDEX statement, your index needs more space. Use the following calculation to determine the approximate number of index pages in this case. Let t be the total index pages on all levels.

$$\text{number of index pages} = (1 + 1/f) \times t$$

Leaf pages: To calculate the number of leaf pages, use the following formula:

$$\text{CEILING}(\text{number of table rows} / \text{entries per page})$$

Nonleaf pages: If there is more than one leaf page, there must be at least one level of nonleaf pages. Each entry on a nonleaf page has an index key value and one pointer, and the nonleaf page is not divided into subpages. Therefore, you can calculate the number of entries per nonleaf page on any level using the following formula:

$$\text{FLOOR}((s \times (4050-3)) / (k+3)) + 1$$

To determine the number of nonleaf pages per index level, apply the following formula:

$$\text{CEILING}(\text{pages on lower level} / \text{entries per nonleaf page})$$

Reuse the above formula until a nonleaf level with only one page is found. This page represents the index root page.

The total space requirement: Finally, the number of kilobytes required for an index built by LOAD is $4 \times (p+2)$, where p is the total number of pages required for all levels of the index.

For an example of the entire calculation, assume that an index is to be defined with these characteristics:

- It is unique.
- The table it indexes has 100 000 rows.
- The key is a single column defined as CHAR(10) NOT NULL.
- The index has one subpage for each leaf page.

The calculations are shown in Table 22 on page 2-32.

A Unique Index with One Subpage per Page:

$$\text{Entries per page} \cong \text{FLOOR}(s \times 4050 / (k + 4))$$

A Unique Index with More Than One Subpage per Page: Calculate as follows:

$$\text{Entries per page} \cong m \times \text{FLOOR}(s \times \text{FLOOR}(4067 / m - k - 21) / (k + 4))$$

A Nonunique Index with One Subpage per Page: Calculate as follows:

$$\text{Entries per page} \cong n \times \text{FLOOR}(s \times 4050 / (k + (4 \times n) + 6))$$

If the calculated number of entries per page is zero, then use:

$$\text{Entries per page} \cong \text{FLOOR}(s \times (4044 - k)) / 4$$

A Nonunique Index with More Than One Subpage per Page: Calculate as follows:

$$\text{Entries per page} \cong n \times m \times \text{FLOOR}(s \times \text{FLOOR}(4067 / m - k - 21) / (k + (4 \times n) + 6))$$

If the calculated number of entries per page is zero, then use:

$$\text{Entries per page} \cong m \times \text{FLOOR}(s \times \text{FLOOR}(4067 / m - k - 21) - (k + 6)) / 4$$

# Table 22. The Total Space Requirement for a Unique Type 1 Index		
Quantity	Calculation	Result
Length of key	(k)	10
Proportion of available space	(s): (100-10)/100	0.9
Number of subpages per page	m	1
Average number of duplicate keys	n	1
Number of entries per page	FLOOR(4050 × 0.9/(10+4))	260
Number of entries per nonleaf page	FLOOR(.9×(4050-3)/(10+3))+1	281
Number of leaf pages	CEILING(100000/260)	385
Number of nonleaf pages on second level	CEILING(385/281)	2
Number of nonleaf pages on third level	CEILING(2/281)	1
Total index pages on all levels	385 + 2 + 1	388
TOTAL SPACE REQUIRED, in kilobytes	4 × (388 + 2)	1560

Type 2 Index Calculations

Use the same variables and operations as for type 1 indexes.

Calculate Pages for a Unique Index: Use the following calculations to estimate the number of leaf and nonleaf pages in a unique index.

Calculate the *total leaf pages*:

- #
1. *Space per key* $\cong k + r + 3$
 2. *Usable space per page* $\cong \text{FLOOR}((100 - f) \times 4038 / 100)$
 3. *Entries per page* $\cong \text{FLOOR}(\text{usable space per page} / \text{space per key})$
 4. **Total leaf pages** $\cong \text{CEILING}(\text{number of table rows} / \text{entries per page})$

Calculate the *total nonleaf pages*:

1. *Space per key* $\cong k + 7$
2. *Usable space per page* $\cong \text{FLOOR}(\text{MAX}(90, (100 - f)) \times 4046 / 100)$
3. *Entries per page* $\cong \text{FLOOR}(\text{usable space per page} / \text{space per key})$
4. *Minimum child pages* $\cong \text{MAX}(2, (\text{entries per page} + 1))$
5. *Level 2 pages* $\cong \text{CEILING}(\text{total leaf pages} / \text{minimum child pages})$
6. *Level 3 pages* $\cong \text{CEILING}(\text{level 2 pages} / \text{minimum child pages})$
7. *Level x pages* $\cong \text{CEILING}(\text{previous level pages} / \text{minimum child pages})$
8. **Total nonleaf pages** $\cong (\text{level 2 pages} + \text{level 3 pages} + \dots + \text{level x pages})$ until the number of level x pages = 1)

Calculate Pages for a Nonunique Index: Use the following calculations to estimate the number of leaf and nonleaf pages for a nonunique index.

Calculate the *total leaf pages*:

- #
1. *Space per key* $\cong 4 + k + (n \times (r+1))$
 2. *Usable space per page* $\cong \text{FLOOR}((100 - f) \times 4038 / 100)$
 3. *Key entries per page* $\cong n \times (\text{usable space per page} / \text{space per key})$
 4. *Remaining space per page* $\cong \text{usable space per page} - (\text{key entries per page} / n) \times \text{space per key}$

5. *Data records per partial entry* \cong FLOOR((remaining space per page - (k + 4)) / 5)
6. *Partial entries per page* \cong (n / CEILING(n / data records per partial entry)) if data records per partial entry \geq 1, or 0 if data records per partial entry $<$ 1
7. *Entries per page* \cong MAX(1, (key entries per page + partial entries per page))
8. **Total leaf pages** \cong CEILING(number of table rows / entries per page)

Calculate the *total nonleaf pages*:

#

1. *Space per key* \cong k + r + 7
2. *Usable space per page* \cong FLOOR (MAX(90, (100- f)) \times (4046/100))
3. *Entries per page* \cong FLOOR((usable space per page / space per key))
4. *Minimum child pages* \cong MAX(2, (entries per page + 1))
5. *Level 2 pages* \cong CEILING(total leaf pages / minimum child pages)
6. *Level 3 pages* \cong CEILING(level 2 pages / minimum child pages)
7. *Level x pages* \cong CEILING(previous level pages / minimum child pages)
8. **Total nonleaf pages** \cong (level 2 pages + level 3 pages + ...+ level x pages until x = 1)

Calculate the Total Space Requirement: Finally, calculate the number of kilobytes required for an index built by LOAD.

1. *Free pages* \cong FLOOR(total leaf pages / p), or 0 if p = 0
2. *Space map pages* \cong CEILING((tree pages + free pages) / 8131)
3. *Tree pages* \cong MAX(2, (total leaf pages + total nonleaf pages))
4. *Total index pages* \cong MAX(4, (1 + tree pages + free pages + space map pages))
5. *Total space requirement* \cong 4 \times (total index pages + 2)

In the following example of the entire calculation, assume that an index is defined with these characteristics:

- It is unique.
- The table it indexes has 100 000 rows.
- The key is a single column defined as CHAR(10) NOT NULL.
- The value of PCTFREE is 5.
- The value of FREEPAGE is 4.

The calculations are shown in Table 23 on page 2-34.

Table 23. The Total Space Requirement for a Type 2 Index

Quantity	Calculation	Result
Length of key	k	10
Average number of duplicate keys	n	1
PCTFREE	f	5
FREEPAGE	p	4
Calculate Total Leaf Pages		
Space per key	$k + 7$	17
Usable space per page	$\text{FLOOR}((100 - f) \times 4038 / 100)$	3836
Entries per page	$\text{FLOOR}(\text{usable space per page} / \text{space per key})$	225
Total leaf pages	$\text{CEILING}(\text{number of table rows} / \text{entries per page})$	445
Calculate Total Nonleaf Pages		
Space per key	$k + 7$	17
Usable space per page	$\text{FLOOR}(\text{MAX}(90, (100 - f)) \times (4046/100))$	3844
Entries per page	$\text{FLOOR}(\text{usable space per page} / \text{space per key})$	226
Minimum child pages	$\text{MAX}(2, (\text{entries per page} + 1))$	227
Level 2 pages	$\text{CEILING}(\text{total leaf pages} / \text{minimum child pages})$	2
Level 3 pages	$\text{CEILING}(\text{level 2 pages} / \text{minimum child pages})$	1
Total nonleaf pages	$(\text{level 2 pages} + \text{level 3 pages} + \dots + \text{level } x \text{ pages until } x = 1)$	3
Calculate Total Space Required		
Free pages	$\text{FLOOR}(\text{total leaf pages} / p)$, or 0 if $p = 0$	111
Tree pages	$\text{MAX}(2, (\text{total leaf pages} + \text{total nonleaf pages}))$	448
Space map pages	$\text{CEILING}((\text{tree pages} + \text{free pages}) / 8131)$	1
Total index pages	$\text{MAX}(4, (1 + \text{tree pages} + \text{free pages} + \text{space map pages}))$	561
TOTAL SPACE REQUIRED, in kilobytes	$4 \times (\text{total index pages} + 2)$	2252

Virtual Storage for Address Spaces

DB2 uses six private address spaces:

- DB2 distributed data facility address space (DSN1DIST)
- IRLM address space (IRLMPROC)
- DB2 database services address space (DSN1DBM1)
- DB2 system services address space (DSN1MSTR)
- DB2 allied agent address space
- DB2 stored procedures address space (DSN1SPAS)

DB2 also uses extended common service area (ECSA).

You might notice that the samples jobs sometimes use a region size of 0K. This region size is meant to simplify the installation process in those particular cases. In the following sections there are some recommendations from DB2 on region sizes. These recommendations are based on 'average' use under 'normal' circumstances on 'typical' systems. Your requirements might be quite different.

DB2 Distributed Data Facility Address Space (DSN1DIST)

This address space supports VTAM communications with other DB2 subsystems and execution of database access requests on behalf of remote users. We recommend you use the default region size of 0KB . This address space is started as part of DDF initialization. The start up procedure is DSN1DIST.

IRLM Address Space (IRLMPROC)

DB2 uses the IRLM to manage locks. When you install DB2, if you specify NO for the CROSS MEMORY option on installation panel DSNTIPJ, or if you specify PC=NO on the START IRLMPROC command, the IRLM control block structures relating to locking are in the extended common service area (ECSA). In this case, the amount of storage available to IRLM is limited by the value you specify for MAXIMUM ECSA. The IBM-supplied default value for MAXIMUM ECSA is 6MB .

If you specify YES for CROSS MEMORY on installation panel DSNTIPJ, or if you specify PC=YES on the START IRLMPROC command, IRLM places its control block structures relating to locking in the IRLM private address space.

When row level locking is used, the number of locks acquired by DB2 might increase, which might in turn increase the amount of storage required by IRLM. The number of locks that are acquired is dependent on your application. You can estimate the IRLM control block structure at 250 bytes per lock. First, plan 6MB for the IRLM control block structure, then adjust according to your needs. Monitor DB2 lock use and processor use at your site. Adjust the installation parameter values for IRLM after you gain some experience with DB2.

You can adjust the MAXCSA parameter dynamically using the MODIFY
irlmproc,SET,CSA command. The new value remains in effect until the next time
IRLM is stopped and restarted.

Enabling data sharing further increases storage required by IRLM. Sysplex query parallelism requires additional storage beyond that required for data sharing.

For additional information on the IRLM startup procedure parameters, see
Command Reference.

DB2 System Services Address Space (DSN1MSTR)

The DB2 system services address space needs less space than the database services address space. We recommend you specify 5000KB for the system services address space, but plan to use 2MB. The default start up procedure is DSN1MSTR.

DB2 Database Services Address Space (DSN1DBM1)

The DB2 database services address space is the largest DB2 address space. The default start up procedure is DSN1DBM1. First, plan for a minimum of 14MB in this address space, with 1334KB below the 16MB line. The rest of this chapter discusses the various components of the database services address space. For more information, see "Virtual Storage for Storage Pools and Working Storage" on page 2-38.

Most modules, control blocks, and buffers reside in the extended private area. A DB2 subsystem with 200 concurrent users and 2000 open data sets should need less than 2MB of virtual storage below the 16MB line.

Allied Agent Address Space

DB2 refers to the user address spaces as the allied agent address space. This can include TSO, IMS, CICS, and batch address spaces. The size of the DB2 attach code in the allied agent address space depends on which attachment facilities you use. TSO requires about 130KB for the DSN command. CAF and IMS require about 36KB for the DB2 attach code. For all attachment facilities, except CICS Version 4, the DB2 attach code must run below the 16MB line of virtual storage. Applications can run above the 16MB line. To calculate space requirements for the CICS attachment facility, see “Calculating Space Requirements for the CICS Attachment Facility” on page 2-381.

Common Service Area

Some of the DB2 load modules and control blocks are in common storage. Most of the space is in the extended common service area (ECSA). With few exceptions, the CSA-resident load modules are link-edited with the residency attribute of RMODE(ANY). Most of the modules reside in ECSA (above the 16MB line of virtual storage), as do most of the global control blocks, including the IRLM control blocks.

The residual requirement for CSA (below the 16MB line) is expected to be less than 40KB for each DB2 subsystem and 48KB for each IRLM started.

To figure out how much storage is needed for ECSA (above the 16MB line) for each DB2 subsystem, follow these guidelines:

- Start with 2.1MB of ECSA for the base and the first 100 users
- If you specify NO for the CROSS MEMORY option on installation panel DSNTIPJ, add the MAXIMUM ECSA option on installation panel DSNTIPJ (the default is 6MB)
- Add 4KB for each additional user
- Add 3KB for each active remote thread
- Add up to 4MB for instrumentation facility interface (IFI) buffers as requested by the monitoring programs

Specify this sum or a value larger than this sum as the second value of the CSA parameter of the IEASYSxx MVS SYS1.PARMLIB member. It is better to specify values that are too high rather than too low; making your values too low can result in your having to IPL MVS. For example, if the ECSA size is too small, MVS places DB2's global load modules and control blocks in CSA below the 16MB line instead of above it. This can cause problems with coexisting MVS subsystems.

Monitor your use of CSA and ECSA, and increase those values if necessary. Monitoring CSA below the 16MB line can indicate whether or not you need to increase the size of the ECSA.

When you IPL MVS, you can override the CSA size with this syntax:

CSA=(a,b)

where:

- *a* is the number of kilobytes of CSA storage below the 16MB line
- *b* is the number of kilobytes of CSA storage above the 16MB line

These values are rounded down (CSA) or up (ECSA) to the next 1MB boundary. For more information, see *OS/390 MVS Initialization and Tuning Guide*.

DB2-Established Stored Procedures Address Space (DSN1SPAS)

This is a DB2-established address space that provides an isolated environment in which to execute stored procedures. Once all members of a data sharing group are migrated and fallback is unlikely, use WLM-established stored procedures address spaces only. This DB2-established stored procedures address space is provided for compatibility while you migrate.

We recommend using partitioned data set extended (PDSE) for load libraries containing stored procedures. Using PDSEs may eliminate your need to stop and start the stored procedures address space due to growth of the load libraries. If a load library grows from additions or replacements, the library may have to be extended.

If you use PDSEs for the load libraries, the new extent information is dynamically updated and you do not need to stop and start the address space. If PDSs are used, load failures may occur because the new extent information is not available.

See Section 6 of *Application Programming and SQL Guide* for more information.

WLM-Established Stored Procedures Address Spaces

These are WLM-established address spaces that provide multiple isolated environments for stored procedures. The advantages of using WLM-established stored procedures address spaces over a DB2-established stored procedures address space include:

- Stored procedures are isolated so failures do not affect other stored procedures.
- Reduced demand for storage below the 16MB line, removing the limitation on the number of stored procedures that can run concurrently.
- Stored procedures inherit the MVS dispatching priority of the DB2 thread that issues the CALL statement.

Each WLM-established stored procedures address space is associated with a Workload Manager environment.

DB2 for OS/390 stored procedures support main and sub programs, which requires additional storage per TCB. However, because you can run fewer programs in an address space, you can use less storage below the 16MB line in each address space. For more information see Section 5 (Volume 2) of *Administration Guide*.

Virtual Storage for Storage Pools and Working Storage

You specify values during the ISPF tailoring session that the DSNTINST CLIST uses to calculate main storage size. We recommend that you determine these values based upon your estimated application workload before you install or migrate DB2.

These values provide an estimate of the private area needed by DB2 DSN1DBM1 address space, the largest of the DB2 address spaces. If the estimated virtual storage for the address space is not available, you can reevaluate the sizes you requested. The size of virtual storage for an address space can not exceed 2GB.

Data compression users have an additional consideration: the amount of DSN1DBM1 storage used by the compression dictionary. This consideration is addressed in “Calculating the Space Required for a Dictionary” on page 2-28.

These calculations are planning estimates. The values noted do not provide the exact limits, but indicate a reasonable range of values. More detailed information is provided in Section 5 (Volume 2) of *Administration Guide*.

This section presents information about the values used to calculate region size:

- Buffer pool size
- Sort pool size
- Record identifier (RID) pool size
- Environmental descriptor manager (EDM) pool size
- VSAM data set control block storage size
- Working storage size

The CLIST adds a fixed code size to the sum of these values to determine the main storage size.

Of these values, the sum of the buffer pool size, sort pool size, RID pool size, EDM pool size, data set control block storage size (for your table spaces and indexes), and working storage size must fit the region size that is permitted for DB2. This sum is important because most of the space is allocated above the 16MB line of virtual storage.

After you specify the values listed above, the CLIST calculates the EDM pool size and the size needed for the data set control blocks. The CLIST adds in the buffer pool size, the sort pool size, the RID pool size, the working storage size, and the fixed code size to update the region size used in the DB2 start procedures. The CLIST also displays this information on installation panel DSNTIPC, which is described on page 2-215.

Use the formulas in this section to estimate your storage needs. For your reference, the default values are included where appropriate.

Buffer Pool Size Calculation

Buffer pools are areas of virtual storage used to satisfy the buffering requirements for one or more table spaces or indexes. All DB2s use virtual buffer pools, backed by central storage, expanded storage, or auxiliary storage. If your system meets the requirements, you can use a second level of storage known as hiperpools.

Hiperpools are extensions to virtual buffer pools that exist in expanded storage known as hiperspace.

Virtual Buffer Pools: For best results, use at least 40KB of buffer pool space for each concurrent user. We recommend a value of 60KB or more for improved performance. Very simple SQL statements accessing small amounts of data can require less than this. Complex SQL statements accessing large amounts of data can require more than this amount.

During installation, you can set the sizes on the install panels. You can use the command ALTER BUFFERPOOL to later alter the sizes and other attributes of up to fifty buffer pools for 4KB page sets and ten buffer pools for 32K table spaces. The ALTER BUFFERPOOL command can make the changes dynamically while DB2 is up. See Section 5 (Volume 2) of *Administration Guide* for information on changing the buffer pool sizes.

Use Table 24 to calculate the virtual buffer pool sizes for your subsystem.

Virtual Buffer Pool Calculation		Default
Buffers for BP0	_____ × 4KB = _____	2000 × 4KB = 8000KB
Buffers for BP1	+ _____ × 4KB = _____	+ 0 × 4KB = 0KB
Buffers for BP2	+ _____ × 4KB = _____	+ 0 × 4KB = 0KB
.	.	.
Buffers for BP49	+ _____ × 4KB = _____	+ 0 × 4KB = 0KB
Buffers for BP32K	+ _____ × 32KB = _____ = _____	+24 × 32KB = 768KB = 8768KB
.	.	.
Buffers for BP32K9	+ _____ × 32KB = _____ = _____	+ 0 × 32KB = 0KB = 8768KB

Hiperpools: A hiperpool is an extension to a virtual buffer pool. It is a second level cache using expanded storage for data that is not accessed frequently enough to stay in the virtual buffer pool. You can use hiperpools if your system meets the requirements described in “Buffer Pools” on page 1-29.

During installation, you specify whether you want a hiperpool to exist for a given virtual buffer pool by specifying a size in the hiperpool field for that virtual buffer pool field. A hiperpool cannot exist unless it is associated with a virtual buffer pool. See “Buffer pool Sizes Panel 1: DSNTIP1” on page 2-150 and “Buffer pool Sizes Panel 2: DSNTIP2” on page 2-152 for details about the installation options for hiperpool sizes. Again, you are not committed to the decisions you make during installation—you can change the size of a hiperpool any time with the ALTER BUFFERPOOL command, even while DB2 is running.

DB2 can use up to 8GB of expanded storage for hiperpools. Hiperpools can span up to four 2GB expanded storage areas known as hiperspaces. Hiperpools fully use a 2GB hiperspace before creating another one. For each 2GB used by data in hiperpools, DB2 needs about 28MB of central virtual storage. Therefore, the most central virtual storage you need to support hiperpools, even if you used them to capacity, would be about 112MB.

Sort Pool Size Calculation

DB2's sort process uses two kinds of storage: local storage and buffer pool storage.

Sort storage in local storage: The sort process creates fixed length storage pools in local storage for internal sort structures and work areas. Local storage is created above the 16MB line at allocation time.

The DB2 sort work area (in-memory) has the following storage boundaries for **each** concurrent sort user:

Minimum sort storage = 240KB
Maximum sort storage = 64MB

The size of the sort pool is calculated by the installation CLIST, but you can override the calculated value for sort pool size by entering the desired size on installation panel DSNTIPC. Estimate the storage required for a sort pool with the following formula:

$$16000 \times (12 + \text{sort key length} + \text{sort data length} + 4 \text{ (if ESA hardware sort assist)})$$

See Section 5 (Volume 2) of *Administration Guide* for instructions on choosing sizes for optimal performance.

Storage for sort processing is acquired incrementally in 4KB blocks, until it reaches the maximum limit. If sort reaches its limit, an abend occurs. If this happens, increase the total buffer pool size so that more DASD space is allocated for sort activity.

Sort storage in the DB2 buffer pools: Sort processing uses pages in the DB2 buffer pool for its initial input, for work files that contain intermediate results, and for the final output.

The buffer pools are not dedicated to sort work files; the amount of sort activity determines how much the buffer pools are used. If there is heavy sort activity, sort records that have been written to the work files are temporarily written to DASD until buffer pool space becomes available.

The buffers that are used for work files are considered as sequentially accessed pages. The percentage of the buffer pool used for work files can be adjusted by using the *VPSEQT* parameter of the command —ALTER BUFFERPOOL. If a buffer pool is used only for work files, you might set *VPSEQT* to 100%. If there is not enough storage allocated to complete sort processing, you must allocate more DASD space for the work file database. For more information, see “DASD Requirements for the Work File Database” on page 2-21.

RID Pool Size Calculation

The RID pool is an area of local storage reserved for record identifier (RID) sort processing, including RID list sort processing.

The RID pool is created at start up time, but no space is allocated until RID storage is needed. When RID storage is needed, it is allocated above the 16MB line in 16KB blocks known as RID blocks.

Keep the following amounts in mind when calculating RID pool size:

Startup RID storage = 0 (but acquired as needed in 16KB blocks)

Maximum RID storage = 1000MB

You can override the calculated value for RID pool size by entering the desired size on installation panel DSNTIPC. If you do this, estimate the storage required for the RID pool with the following formula:

$$\text{number of concurrent RID processing activities} \times \text{average number of RIDs} \times 2 \\ \times 5 \text{ (bytes per RID)}$$

See Section 5 (Volume 2) of *Administration Guide* for an example of changing a RID pool size to improve performance.

EDM Pool Size Calculation

The environmental descriptor manager (EDM) pool contains active and skeleton application plans and packages, as well as database descriptors. This section describes how to estimate the space needed for plans (including packages) and database descriptors. These estimations do not account for every factor that affects EDM pool size—your actual EDM pool size will vary. Generally, we recommend that the EDM pool be at least 10 times the size of the largest database descriptor or plan, whichever is greater.

The CLIST generates the EDM pool value. This value is on installation panel DSNTIPC, which is described on page 2-215.

Calculating the EDM Pool Space Needed for Plans and Packages

The first part of the EDM pool calculation involves the space for plans and packages. Plans and packages are very different, but they are treated similarly for storage planning. To estimate the EDM pool space needed for plans, use the following variables:

- Let *concpans* be the number of concurrently executing plans. This is the value specified as MAX USERS on installation panel DSNTIPE.
- Let *maxplans* be the maximum number of unique plans you want in the EDM pool at any given time. Estimate this by taking one fourth of the total plans you have.
- Let *statsize* be the average statement size. To calculate the average statement size, add 1.4KB for each single table statement and 0.2KB for each additional table in the statement. For example, if you estimate one table for each statement, multiply the number of SQL statements by 1.4KB; if you estimate three tables per statement, multiply the number of statements by 1.8KB (1.4KB + 0.2KB + 0.2KB).
- Let *statexec* be the average number of statements executed. The CLIST uses the value in EXECUTED STMTS on installation panel DSNTIPD.

Then calculate as follows:

$$(concplans + maxplans) \times (statsize \times statexec)$$

#

The average plan size changes in proportion to the increase in the number and complexity of SQL statements. The complexity of the access path for SQL statements can also affect plan size.

To calculate your average plan size, you need to allow 1KB to 4KB for control blocks for caching, depending on your BIND option. Let *cntrlblk* be the number of kilobytes allocated for caching. Then, calculate the average plan size with the following formula:

$$(statsize \times statexec) + cntrlblk$$

If you have accepted the defaults, the CLIST makes the following calculations. 1.4KB is added for the single table and 0.2KB for the additional table, making *statsize* = 1.6KB. 1.6KB is multiplied by 15, the default for *statexec*. 1KB is added to the product, and the result is 25KB.

The size of a QMF plan during execution is typically 8KB.

The ACQUIRE option of the BIND command affects the plan size. ACQUIRE(ALLOCATE) results in a larger plan size than ACQUIRE(USE), thus affecting the amount of EDM pool storage the plan consumes. ACQUIRE(ALLOCATE) causes items to be stored with the plan to enhance performance.

#

The RELEASE option of the BIND command can also affect the plan size. RELEASE(DEALLOCATE) can result in a larger plan size than RELEASE(COMMIT), thus affecting the amount of EDM pool storage that the plan consumes. RELEASE(DEALLOCATE) causes items to remain stored with the plan to enhance performance.

#

The increase in plan size is determined by the number of database objects (databases, table spaces, and tables) used by the plan. In CICS and IMS environments, thread reuse tends to increase the number of database objects used by the plan when the RELEASE(DEALLOCATE) BIND option is used.

General-use Programming Interface

After you bind a plan or package, you can check the size by listing the AVGSIZE, PLSIZE or PKSIZE, and CACHESIZE in the SYSPLAN or SYSPACKAGE catalog table. PLSIZE or PKSIZE is the size of the base segment of the plan or package. AVGSIZE is the average size per section. CACHESIZE is the cache size you specify for the authorization ID cache for the plan. To find the plan or package sizes and the size of the authorization ID cache, use the appropriate SQL statement:

For a plan:

```
SELECT NAME, PLSIZE, AVGSIZE, CACHESIZE
       FROM SYSIBM.SYSPLAN
       ORDER BY NAME;
```

For a package:

```

SELECT NAME, PKSIZE, AVGSIZE
FROM SYSIBM.SYSPACKAGE
ORDER BY NAME;

```

To find the number of sections for each DBRM bound with each plan, use the following SQL statement. Add the number of sections per DBRM by plan to find the number of sections per plan.

```

SELECT PLNAME, NAME, MAX(SECTNO)
FROM SYSIBM.SYSSTMT
GROUP BY PLNAME, NAME
ORDER BY PLNAME, NAME;

```

To find the number of sections for each DBRM bound with each package, use the following SQL statement. Add the number of sections per DBRM by package to find the number of sections per package.

```

SELECT COLLID, NAME, VERSION, MAX(SECTNO)
FROM SYSIBM.SYSPACKSTMT
GROUP BY COLLID, NAME, VERSION
ORDER BY COLLID, NAME, VERSION

```

|_____ End of General-use Programming Interface _____|

You can also use similar statements with WHERE clauses to specify the plans or packages you want. The number of sections executed per plan or package can be estimated only from the execution of the application that uses the particular plan or package. Consequently, the amount of storage needed during execution of the application is the base segment size of the plan or package plus the size for the sections being executed.

The storage needed for a plan is the sum of the base size and the size of the executed sections. The storage needed for a package is the sum of the base size, the base size of the package, and the size of the executed sections.

Calculating the EDM Pool Space for the Prepared Statement Cache

If you choose to enable the prepared statement cache for dynamic SQL programs, you must increase the size of your EDM pool or the performance of static plans and packages can be adversely affected. When you use the cache, prepared statements are stored in the EDM pool, the same as static SQL. The number of prepared statements that are stored in the cache depends on the characteristics of the dynamic SQL your application executes. One type typically benefits from caching prepared statements, while the other type usually does not.

The first type of applications use dynamic SQL that is embedded in an application and is used repeatedly. Applications with this type of SQL benefit most from caching prepared statements because the statement can be used from the cache.

On the other hand, applications that contain SQL statements that are seldom used pay the cost of being added to the cache with few benefits. For example, queries from QMF are likely to be prepared and executed only once. Caching prepared statements does not benefit applications that extensively use this kind of SQL.

Estimate the storage needed for the prepared statement cache using these variables.

- Let *maxdynplans* be the maximum number of unique plans containing embedded dynamic SQL that you expect to be running in the system at any given time.
- Let *dynstatsize* be the average statement size of a prepared statement in the cache. To calculate the average prepared statement size, add 1.8KB for each single table statement and 0.2KB for each additional table in the statement. For example, if you assume an average of three tables per statement, the average statement size is 2KB (1.6KB + 0.2 + 0.2).
- Let *dynstatexec* be the average number of different dynamic SQL statements that are executed by plans containing dynamic SQL that are likely to be used repeatedly.
- Let *adhocexec* be the maximum number of unique SQL statements that are likely to be used only once that are executed by all users at any given time. You can estimate this by multiplying the maximum number of users running ad hoc SQL programs and multiplying by 5.

Then calculate the size needed for the Prepared Statement Cache as follows:

$$(maxdynplans \times (dynstatsize + (dynstatexec)) + (adhocexec) \times dynstatsize)$$

The installation CLIST does not attempt to calculate this when calculating the estimated EDM Pool size because it does not have all the information provided by the variables.

Calculating the EDM Pool Space Needed for Database Descriptors

The final part of the EDM pool calculation involves the space for database descriptors (DBDs). To determine this value, multiply the number of concurrently open databases by the average size of the database descriptor.

The database descriptor size is 12KB for the default values. The database descriptor size depends on the number of table spaces, tables, indexes, columns, partitions, referential constraints, table check constraints, and index keys in the database. The DSNTINST CLIST contains the algorithm for calculating the DBD size. The maximum size of a database descriptor is 25% of the size of the EDM pool. You need to ensure that the EDM pool size is at least four times the estimated size of your largest database descriptor.

Using Table 25, you can estimate DBD size based on an estimate of columns per table and tables per database for your site. Assume:

- The same number of tables as table spaces
- 1 index per table
- 2 partitions per table space
- 3 keys per index
- 2 referential relationships per table

These values are the defaults built into the CLIST and are reasonably typical for databases.

Table 25. DBD Sizes for Ranges of Columns and Tables

Columns per Table	10 Tables per Database	20 Tables per Database	30 Tables per Database	40 Tables per Database	50 Tables per Database
10	12KB	24KB	32KB	44KB	56KB
20	12KB	24KB	36KB	48KB	60KB
30	16KB	28KB	40KB	52KB	64KB
40	16KB	28KB	44KB	56KB	68KB
50	16KB	32KB	44KB	60KB	76KB
75	20KB	36KB	52KB	60KB	88KB
100	20KB	40KB	60KB	80KB	100KB
200	32KB	60KB	88KB	120KB	148KB
300	40KB	80KB	120KB	156KB	196KB

Total EDM Pool Space

Use the following variables to calculate EDM Pool space for plans, packages, dynamic statements, and DBDs: Then, add 50KB for overhead. Finally, multiply the total by 1.25 to estimate fragmentation loss.

- Let *concpans* be the number of concurrently executing plans. This is the value specified as MAX USERS on installation panel DSNTIPE.
- Let *maxplans* be the maximum number of unique plans you want in the EDM pool at any given time. Estimate this by taking one fourth of the total plans you have.
- Let *plansize* be the average plan size.
- Let *concdb* be the number of concurrent databases specified on installation panel DSNTIPD.
- Let *dbdsize* be the DBD size.

Then, calculate as follows:

$$(\text{concpans} + \text{maxplans}) \times \text{plansize} + (\text{concdb} \times \text{dbdsize})$$

Then, add 50KB for overhead, and multiply the total by 1.25 to estimate fragmentation loss.

The default, as calculated by the DSNTINST CLIST, is $(70 + 50) \times 25\text{KB} + (100 \times 12\text{KB})$, or 4200KB. Then, 50KB are added overhead, and the total of 4250KB is multiplied by 1.25 to give the result of 5312KB.

#

Recommendation: If you bind plans or packages using DEGREE=ANY, increase the space in the EDM pool by 50-70%.

Data Set Control Block Storage Size Calculation

To determine the total number of open data sets (DSMAX):

- Let *concdb* be the number of concurrent databases specified on installation panel DSNTIPE.
- Let *tables* be the number of tables per database specified on installation panel DSNTIPD.

- # • Let *indexes* be the number of indexes per table. The installation CLIST sets this variable to 2.
- # • Let *tblspaces* be the number of table spaces per database specified on installation panel DSNTIPD.
- | • Let *partts* be the average number of partitioned table spaces per database.
- | • Let *avgpart* be the average number of partitions per partitioned table space.

Then calculate the number of open data sets with the following formula:

$$concdb \times (((tables \times indexes) + tblspaces) + ((2 \times partts \times avgpart) - (2 \times partts)))$$

You can modify DSMAX by editing job DSNTIJUZ or by changing the DSMAX field in macro DSN6SPRM. The maximum number of concurrently open data sets is:

- # • 32767 if you are running on OS/390 Version 2 Release 6 or later
- # • 10000 if you are running on a level of OS/390 that is earlier than Version 2 Release 6

See Section 5 (Volume 2) of *Administration Guide* for information on tailoring DSMAX for performance. Your DSMAX calculations might be different for LARGE table spaces. Nonpartitioned indexes on a LARGE partitioned table space can have up to 128 data sets.

To calculate the main storage required for your data set control blocks, use the following formula:

$$DSMAX \times 1.8KB$$

The default, as calculated by the DSNTINST CLIST, is $2000 \times 1.8KB$, or 3600KB.

This method of calculation ignores partitioned table spaces and partitioned index spaces. It also assumes that all data sets in the database are open if the database is in use. You could enter a smaller value for the number of concurrent databases if typically only a few of the data sets in a database are opened. The larger the value of DSMAX, the longer CLOSE YES data sets stay open.

- # DB2 recommends that you move the scheduler work area (SWA) above the 16MB line of MVS virtual storage by using JES initialization statements, JES exits, or the SMF exit (IEFUJV). This way, you can save approximately 600 bytes per open data set in virtual storage below the 16MB line and avoid potential storage errors. To determine the amount of storage needed below the 16MB line, use 0.3KB for the multiplication factor if the scheduler work area (SWA) is above the line or 0.9KB for the multiplication factor if the SWA is below the line. The calculations in the CLIST presume that SWA is above the line. See Improving Resource Utilization in Section 5 (Volume 2) of *Administration Guide* for more information.

Working Storage Calculation

Working storage is that portion of main storage, above and below the 16MB line, that DB2 needs in the database services address space to hold data temporarily. To estimate the amount of working storage needed, start with 600KB. Add 40KB for each concurrent DB2 user (*concusers*). This value is specified as MAX USERS on installation panel DSNTIPE. Follow this formula:

$$600\text{KB} + (\text{MAX USERS} + \text{MAX REMOTE ACTIVE}) \times 40 + (\text{MAX REMOTE CONNECTED} - \text{MAX REMOTE ACTIVE}) \times 4$$

The default, as calculated by the DSNTINST CLIST, is:

$$600\text{KB} + (70 + 64) \times 40 + (64 - 64) \times 4, \text{ or } 5960$$

If you use dynamic SQL, you need more working storage and less EDM pool space than if you use static SQL. QMF users have a very small plan in EDM pool, usually 12KB. Users of static SQL have larger plan sizes as noted above, typically varying from 10KB to 80KB. Typical sites would use about 76KB per thread (a structure that describes an application connection to DB2) of working storage for dynamic SQL users, and 24KB per thread for static SQL users. For information about thread creation, see Section 5 (Volume 2) of *Administration Guide*. The CLIST does not include information about open compressed table spaces. Therefore, if you use compressed table spaces, you need additional storage. See “Virtual storage requirements” on page 2-29 for storage information about compressed table spaces.

Virtual Storage Below the 16MB Line

This calculation produces an estimate of virtual storage constraints below the 16MB line in the DB2 database services address space.

Most of the needed virtual storage is in extended private storage, including the buffer pool, the EDM pool, almost all of the code, and a significant amount of working storage. This is the difference between the total storage and the estimated region size. The region size estimate does not include extended private storage—it includes only the data set control block storage size and some code. To estimate the size of storage below the 16MB line, use the following formula:

$$600\text{KB} + \text{MAX USERS} + \text{MAX REMOTE ACTIVE} + (\text{DSMAX} \times 0.3)$$

The default, as calculated by the DSNTINST CLIST, is $600\text{KB} + 70 + 64 + (2000 \times 0.3\text{KB})$, or 1334KB. If the scheduler work area (SWA) is above the 16MB line, multiply the number of data sets by 0.3KB; if the SWA is below the 16MB line, multiply the number of data sets by 0.9KB.

Table 26 shows total main storage calculation. Place your estimates in the spaces provided and make the indicated calculations.

<i>Table 26. Main Storage Size Calculation</i>		
Category	Your Size	Default
EDM pool storage size		7562KB
Buffer pool size	+	+ 8768KB
Sort pool size	+	+ 876KB
RID pool size	+	+ 4384KB
Data set control block storage size	+	+ 3600KB
Code storage size	+ 4300KB	+ 4300KB
Working storage size	+	+ 5960KB
Total main storage size (above 16MB line)	=	= 35450KB
Region Size (below 16MB line) (assume SWA above the line)		1334KB

The CLIST Calculations Panel, DSNTIPC, displays storage sizes calculated by the DSNTINST CLIST. For more information, see “Install DB2—CLIST Calculations Panel 1: DSNTIPC” on page 2-215.

Real Storage

DB2 can use real storage, which includes both central and expanded storage, to reduce I/O and processor times and to improve response time and throughput. The amount of real storage DB2 needs varies greatly. Some customers find that they need several times the estimates listed below, while others need less. The amount of storage is an important parameter in DB2 performance. Performance monitoring programs give you a more accurate estimate of your storage requirements than the formulas in this section.

For the DB2 buffer pools, the EDM pool, and working storage, the amount of real storage must be the same as the amount of virtual storage. Paging activity in the buffers is an indication of a problem. If there is not enough real storage to hold the buffers, then the buffers need to be reduced. This might mean fewer concurrent users. Additional space is needed to contain locks, the working set of code in all address spaces, log buffers, and ECSA and CSA space. Because some of the figures used in virtual storage calculations are maximums, while the real storage figures typically use activity for the peak, changes are needed in the calculations. The virtual storage figures concentrate on the most constrained address space, but real storage work must include them all. For more information on each category, see the pages specified in Table 27 on page 2-49 and Table 28 on page 2-49.

Category	Default	Virtual Size	× Factor	= Real Size	See Page
Buffer pools	8768KB	_____	1.0	_____	Note 1
Sort pool	876KB	_____	0.5	_____	2-40
RID pool	4384KB	_____	0.5	+ _____	2-41
EDM pool	7562KB	_____	1.0	+ _____	2-41
Data set size	3600KB	_____	0.6	+ _____	2-45
Code size + 1100 (4 address spaces)	5400KB	5400KB	0.5	+ 2700KB	
Working storage + (DSCF + DDF)	6960KB	_____	1.0	+ _____	2-46
Log buffers	400KB	_____	0.8	+ _____	Note 1
Lock space	5000KB	_____	0.4	+ _____	Note 1
CSA/ECSA	2160KB	_____	0.4	+ _____	2-36

Total default real storage size =				= _____	

Note: 1 See Section 5 (Volume 2) of *Administration Guide*

Category	Virtual Size	× Factor	= Real Size
Buffer pools	8768KB	1.0	8768KB
Sort pool	876KB	0.5	438KB
RID pool	4384KB	0.5	2192KB
EDM pool	7562KB	1.0	7562KB
Data set size	3600KB	0.6	2160KB
Code size + 1100 (4 address spaces)	5400KB	0.5	2700KB
Working storage + (DSCF + DDF)	6960KB	1.0	6960KB
Log buffers	400KB	0.8	320KB
Lock space	5000KB	0.4	2000KB
CSA/ECSA	2160KB	0.4	864KB

Total real storage size =			= 31714KB

Using rough estimates, Table 29 shows estimates of the amount of additional real storage needed by several kinds of users. If you have more concurrent users, plan to add real storage.

Table 29. Additional Real Storage for More Users

Type of User	Additional Real Storage
Transaction	150KB
Query	400KB
Batch	700KB

Chapter 2-3. Setting Up DB2 Online Help

The online help for DB2 for OS/390 Version 5 lets you select information in an online DB2 book. The new help has an enhanced display and an excellent search capability.

If you choose not to use online help, invoke the DSNTINST CLIST as described in “Invoking the CLIST” on page 2-90. To use online help, complete the setup instructions in this chapter and then invoke the DSNTINS0 CLIST.

If BookManager READ/MVS is not installed at your site: Complete the following tasks:

- “Installing the Online Help Reader” on page 2-52
- “Unloading DB2 Books” on page 2-59
- “Verifying Online Help and Setting Exit Options” on page 2-63
- Concatenate the new DB2 ISPF command table library (*prefix.SDSNSPFT*) in with the ISPTLIB DD statements in your TSO logon procedures and in any of the CLISTs where they might be allocated. For more information on how DB2 uses the command table library, see “Connecting DB2I Panels to the ISPF Main Panel” on page 2-246.

You also need to allocate the DB2 target libraries for books using your TSO logon procedures. This type of allocation is because the online help reader uses BookManager READ/MVS, which does not support the use of LIBDEFs. See APAR PN20713 for an explanation of this restriction.

If BookManager READ/MVS is installed at your site: Do not install the online help reader; it has fewer functions than BookManager READ/MVS. Instead, make sure that you have access to BookManager READ/MVS by entering the TSO command option:

```
BOOKMGR
```

If the command fails, request access from the system programmer who is in charge of BookManager READ/MVS at your site.

If you are setting up DB2 online help on multiple systems, we recommend that you use BookManager READ/MVS on each system. If BookManager READ/MVS is not installed on some systems, you can install the online help reader shipped with the DB2 tape on those systems. However, be aware that the online help reader has fewer functions than READ/MVS, and users might become confused by the differences in the interface.

Complete the following tasks to set up online help:

- “Unloading DB2 Books” on page 2-59
- “Verifying Online Help and Setting Exit Options” on page 2-63
- Concatenate the new DB2 ISPF command table library (*prefix.SDSNSPFT*) in with the ISPTLIB DD statements in your TSO logon procedures and in any of the CLISTs where they might be allocated.

You must have sufficient storage for the online books as described in Table 30 on page 2-52 and Table 31 on page 2-52. If you install the Online Help Reader, additional storage requirements are listed beginning with Table 32 on page 2-53.

Table 30. Storage Requirements for Target Libraries for Books

Data Set Name or Library Name	DSORG	RECFM	LRECL	BLK SIZE	BLOCKS	DIR.
SDSNBKS	PO	FB	4096	28672	125	2
SDSNINDX	PO	FB	4096	28672	100	1
SDSNINST	PO	FB	80	8800	18	2
SDSNSHLF	PO	FB	80	8800	15	1

Table 31. Storage Requirements for Distribution Libraries for Books

Data Set Name or Library Name	DSORG	RECFM	LRECL	BLK SIZE	BLOCKS	DIR.
ADSNBKS	PO	FB	4096	28672	125	2
ADSNINDX	PO	FB	4096	28672	100	1
ADSNINST	PO	FB	80	8800	18	2
ADSNSHLF	PO	FB	80	8800	15	1

Installing the Online Help Reader

Together with the DB2 program tape, you received a DB2 Online Help 9-track magnetic tape, written at 6250 bytes per inch (BPI), a 3480 cartridge, or a 4mm cartridge.

It is recommended that you install the fix for PTF UL99726 or UL99727 for a C/370 Library problem before you install the online help reader. If you do not install this fix, you may experience occasional performance problems using DB2 Online Help.

The POST-APPLY LINK-EDIT step described in “Performing the Post-APPLY Link-Edit Step” on page 2-57 must be rerun when maintenance is applied to ISPF, C/370, LE/370, or GDDM.

Storage Requirements for the Online Help Reader

The equivalent of 90 cylinders of IBM 3390 Direct Access Storage is recommended to hold the target and distribution libraries (including the linguistic dictionaries) that come on your DB2 Online Help distribution tapes. Table 32 on page 2-53 through Table 35 on page 2-54 list the target and distribution libraries (data sets) and their attributes.

Keep in mind that the number of blocks and directory blocks specified is the actual minimum space required by the online help reader after the program is installed and the data sets are compressed. When allocating these data sets, you may want to specify additional space and directory blocks to allow for maintenance.

Table 32. Storage Requirements for Target Libraries for Base Product

Data Set Name or Library Name	DSORG	RECFM	LRECL	BLK SIZE	BLOCKS	DIR.
SEOYLPA	PO	U	0	6144	1008	5
SEOYLOAD	PO	U	0	6144	2300	20
SEOYCLIB	PO	FB	80	6160	500	10
SEOYSAMP	PO	FB	80	6160	110	5
SEOYLEXS	PO	FB	512	4096	9000	5
SEOYPROC	PO	FB	80	6160	16	5

Table 33. Storage Requirements for Target Libraries for U.S. English

Data Set Name or Library Name	DSORG	RECFM	LRECL	BLK SIZE	BLOCKS	DIR.
SEOXFONT	PO	VBM	8300	8304	24	5
SEOYMENU	PO	FB	80	6160	80	20
SEOYPENU	PO	FB	80	6160	2600	185
SEOYTENU	PO	FB	80	6160	40	5
SEOYSENU	PO	VB	259	6475	8	5
SEOYXENU	PO	FB	4096	8192	170	5
SEOXOENU	PO	FB	4096	8192	240	5
SEOYLENU	PO	FB	80	6160	8	5
SEOYIENU	PO	VB	8300	8304	16	5
SEOX1ENU	PO	VB	259	6220	100	5
SEOX2ENU	PO	VB	259	6220	40	5
SEOX3ENU	PO	VB	259	6220	380	5
SEOX4ENU	PO	VB	259	6220	110	5
SEOYBENU	PO	FB	80	6160	16	5
SEOYGENU	PO	FB	80	6160	16	5
SEOXIENU	PO	VBM	8300	8304	92	5

Table 34. Storage Requirements for Distribution Libraries for Base Product

Data Set Name or Library Name	DSORG	RECFM	LRECL	BLK SIZE	BLOCKS	DIR.
AEOYLOAD	PO	U	0	6144	2300	20
AEOYCLIB	PO	FB	80	6160	500	10
AEOYSAMP	PO	FB	80	6160	110	5
AEOYLEXS	PO	FB	512	4096	9000	5
AEOYPROC	PO	FB	80	6160	16	5

Table 35. Storage Requirements for Distribution Libraries for U.S. English

Data Set Name or Library Name	DSORG	RECFM	LRECL	BLK SIZE	BLOCKS	DIR.
AEOXFONT	PO	VBM	8300	8304	24	5
AEOYMENU	PO	FB	80	6160	80	20
AEOYPENU	PO	FB	80	6160	2600	185
AEOYTENU	PO	FB	80	6160	40	5
AEOYSENU	PO	VB	259	6475	8	5
AEOYXENU	PO	FB	4096	8192	170	5
AEOXOENU	PO	FB	4096	8192	240	5
AEOYLENU	PO	FB	80	6160	8	5
AEOYIENU	PO	VB	8300	8304	16	5
AEOX1ENU	PO	VB	259	6220	100	5
AEOX2ENU	PO	VB	259	6220	40	5
AEOX3ENU	PO	VB	259	6220	380	5
AEOX4ENU	PO	VB	259	6220	110	5
AEOYBENU	PO	FB	80	6160	16	5
AEOYGENU	PO	FB	80	6160	16	5
AEOYCENU	PO	U	0	6144	5	1
AEOXIENU	PO	VBM	8300	8304	92	5

Copying Sample Installation Jobs for the Online Help Reader

Before you can use SMP/E to install the online help reader, you should unload the sample installation jobs from the Base Product tape. To copy the sample jobs from the product tape, edit and submit the JCL shown in Figure 18 on page 2-55. After this job has finished, you can use the remaining sample jobs to install the online help reader on your system.

```

//JOB CARD JOB (XXXXXXXX YOUR JOB CARD INFORMATION GOES HERE
//*****
//*
//* JCL TO UNLOAD SAMPLE JOBS FROM THE PRODUCT TAPE
//*
//* ANY INFORMATION WHICH NEEDS TO BE TAILORED
//* TO YOUR SITE IS DESIGNATED WITH MULTIPLE X'S.
//*
//* THE JCLDSK DATA SET NAME SHOULD BE THE DATA SET
//* TO WHICH YOU WANT THE UNLOADED SAMPLE MEMBERS TO
//* BE COPIED.
//*
//*****
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//JCLTAPE DD DSN=IBM.HDB551A.F1,UNIT=XXXX,VOL=SER=DB551A,
// LABEL=(2,SL),DISP=(OLD,KEEP)
//JCLDSK DD DSN=XXXXXXXX,
// DISP=(NEW,CATLG,DELETE),
// VOL=SER=XXXXXX,UNIT=XXXX,
// DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB),
// SPACE=(6160,(45,40,2))
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
COPY OUTDD=JCLDSK,INDD=JCLTAPE
SELECT MEMBER=DSNACPT
SELECT MEMBER=DSNALA
SELECT MEMBER=DSNALB
SELECT MEMBER=DSNALC
SELECT MEMBER=DSNAPPL
SELECT MEMBER=DSNDDD
SELECT MEMBER=DSNPLINK
SELECT MEMBER=DSNREC
SELECT MEMBER=DSNUNL1
SELECT MEMBER=DSNUNL2

/*

```

Figure 18. JCL to Copy Online Help Reader Sample Jobs to DASD

The sample jobs on your product tape help you to allocate, define, receive, apply, link-edit, and copy online books from partitioned to sequential data sets, and accept the target and distribution libraries for the online help reader. You need to make some changes to these sample jobs to reflect your site's naming conventions for data sets and volumes. After you submit the unload JCL provided above, these jobs are in the data set you identified in the JCLDSK DD statement.

The following installation jobs are provided as samples:

Member Name	Comments
DSNALA	Sample job to ALLOCATE the CSI
DSNALB	Sample job to ALLOCATE SMP/E data sets and initialize the CSI
DSNALC	Sample job to ALLOCATE the reader data sets
DSNREC	Sample SMP/E RECEIVE procedure
DSNAPPL	Sample SMP/E APPLY procedure
DSNPLINK	Sample job to link-edit the online help reader to resolve external references
DSNDDD	SMP/E DDDEF statements for online help
DSNUNL1	Sample job to copy dictionary and bookshelf list
DSNACPT	Sample SMP/E ACCEPT procedure
DSNUNL2	Sample job to copy books, index, and bookshelf

Allocating the CSI Data Sets

A sample job that you can use to allocate these data sets is provided in member DSNALA, which you unloaded previously.

Allocating the SMP/E Data Sets

A sample job that you can use to allocate these data sets is provided in member DSNALB, which you unloaded previously.

Allocating the Online Reader Data Sets

A sample job that you can use to allocate these data sets is provided in member DSNALC, which you unloaded previously.

Defining SMP/E DDDEF Entries

Before you can receive the online help reader data sets, you must define SMP/E DDDEF entries for those data sets. A sample job that you can use to define those entries is provided in member DSNDDD, which you unloaded previously.

Performing the SMP/E RECEIVE Step for the Online Help Reader

You are now ready to do the RECEIVE step on the product tapes. The sample JCL to do this is provided in member DSNREC, which you unloaded previously. This job receives both the Base and the U.S. English FMIDs into SMP/E. If you are receiving the FMIDs one at a time, rather than simultaneously, you must receive the Base product first. You should expect a return code of 00 for this function, and the SMP/E message:

```
GIM2050 GIMSMP PROCESSING COMPLETED-HIGHEST RETURN CODE IS 00
```

Performing the SMP/E APPLY Step

Now that you received the contents of the tapes, you can apply the online help reader modules. A sample job to do the APPLY is provided in member DSNAPPL, which you unloaded previously. There are external references that are not resolved during the APPLY step, but are resolved when the POST-APPLY LINK-EDIT step is performed. This SMP/E APPLY procedure should complete with a return code of 04. The linkage editor displays the following message for each load module:

```
IEW0461 WARNING - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL
REFERENCE: NCAL WAS SPECIFIED, OR THE REFERENCE WAS
MARKED FOR RESTRICTED NO-CALL OR NEVERCALL.
```

In an SMS environment, message GIM23903W is displayed.

Performing the Post-APPLY Link-Edit Step

Some link-editing needs to be done outside SMP/E JCLIN processing. A sample job to do this step is provided in member DSNPLINK, which you unloaded previously. This job takes the partially linked load modules in the target library, and links the rest of the necessary modules with them to create an executable program library.

GDDM

GDDM, Version 3.1 or later, enables you to view pictures within an online book. If you are using the online help reader without GDDM, be sure to follow the instructions in the job for resolving GDDM references.

The POST-APPLY LINK-EDIT step should be run any time maintenance is performed on ISPF, the C/370 Library, LE/370, or GDDM.

Copying the Online Dictionary and Bookshelf List

The online dictionary and bookshelf list are shipped as partitioned data sets, but must be copied to sequential data sets before they can be used. Submit the sample JCL DSNUNL1, which you unloaded previously, to copy the dictionary and bookshelf list to a sequential data set.

Performing the SMP/E ACCEPT Step

When you are ready to accept the product permanently into your distribution libraries, use the sample job provided in member DSNACPT, which you unloaded previously. The ACCEPT step copies the BookManager READ/MVS data sets from the distribution tapes (or RELFILES) into your permanent distribution libraries. You should receive a return code of 00 from the ACCEPT function.

Additional Installation Tasks

After completing SMP/E processing, you must complete the following tasks to make the online help reader available to users.

Update the EOXVOPTS Member

Within the EOXVOPTS member of the SEOYCLIB target library, you can list GDDM data sets. GDDM enables you to view pictures within the online help reader. If your site has GDDM installed, list the data set name of the GDDM symbol sets in the QGDDMSYM option. For example:

```
QGDDMSYM = "'SYS1.GDDM.GDDMSYM' "
```

Make Online Help Available to Users

After you have installed the online help reader, you must perform the following tasks:

- Add the SEOYLPA target library to PLPA or the LINKLIST. Do this by adding the SEOYLPA target library to either the LPALSTxx or LNKLSTxx concatenation in SYS1.PARMLIB.
- Add the SEOYLOAD target library to the LINKLIST. Its members should not be placed in PLPA. You should add the SEOYLOAD target library to the SYS1.PARMLIB member LNKLSTxx concatenation.

For more information, see the descriptions of members LPALSTxx and LNKLSTxx in *MVS/ESA Initialization and Tuning Guide*.

It is recommended that the load modules in the SEOYLPA target library reside in PLPA. This creates one copy of BookManager READ in memory for all users to access, thus reducing the amount of real storage required. Since these load modules are marked with RMODE(ANY), they will reside in extended PLPA (that is, above the 16 megabyte line). These modules need approximately 1 megabyte of virtual storage.

If you decide to copy the members of the SEOYLOAD target library to an existing LINKLIST data set, it is only necessary to copy EOYBOOK. Modules EOXMBMG1 and EOYBOOK1 are only used in SMP/E processing and the POST LINK-EDIT step, and modules EOXLCDE and EOXLCADR are only used in SMP/E processing.

After the SEOYLPA target library has been added to the LPALSTxx member, you must perform an initial program load (IPL). Table 36 lists the approximate size of each load module.

Table 36. Approximate Size of the Online Help Reader Load Modules

Module Name	Size
EOXMBMGR	1MB
EOXLLENU	10KB
EOXVX01	1KB
EOXVX02	1KB
EOXVX03	1KB
EOYBOOK	300KB

You also need to make the online help reader panel libraries, tables, messages, and REXX EXECs available to users. Table 37 shows the names of the default libraries that must be concatenated to the ISPF DDNAMEs in the TSO logon procedure, any user-supplied ISPF start-up CLISTs, and language libraries in the STEPLIB that are not on the LINKLST. The online help reader libraries are listed by the default target names; if you change them for your installation, remember to use the new names. Also, as an option, you could copy the members into the appropriate libraries that you are using for ISPF.

Table 37. Online Help Reader Libraries to be Concatenated

Read Data Set Name (default)	DDNAME
DSNHHELP.SEOYCLIB	SYSPROC (or SYSEXEC)
DSNHHELP.SEOYPENU	ISPPLIB
DSNHHELP.SEOYTENU	ISPTLIB
DSNHHELP.SEOYMENU	ISPMLIB

Concatenate the new DB2 ISPF command table library (*prefix.SDSNSPFT*) in with the ISPTLIB DD statements in your TSO logon procedures and in any of the CLISTs where they might be allocated.

Unloading DB2 Books

When you start the installation CLIST, you see a list of books that are loaded specifically for online help. The books that are loaded are:

- Application Programming and SQL Guide
- Command Reference
- Installation Guide
- Utility Guide and Reference

box

If You Installed the Online Help Reader: The sample job to copy the online books, index, and bookshelf from partitioned to sequential data sets is available in member DSNUNL2 on the DB2 Online Help tape.

If BookManager READ/MVS is Installed at Your Site: You must first unload the book data sets from the online help tape, and then unload the books from partitioned to sequential data sets. Complete the following steps:

1. Copy the sample jobs off the tape using the JCL shown in “Copying Sample Installation Jobs for the Online Help Reader” on page 2-54.
2. Modify sample job DSNALA for your site. This job allocates the SMP/E CSI.
3. Modify sample job DSNALB for your site. This job defines SMP/E data sets and initialize the CSI.
4. Sample job DSNALC allocates book data sets. Modify DSNALC for your site, and comment out all target and distribution libraries except the following:

```

//*****
//*                TARGET LIBRARIES
//*****
//SDSNINST DD DSN=&HIQUAL;.SDSNINST,DISP=(NEW,CATLG),
//          UNIT=&TUNIT;,VOL=SER=&TVOL;;
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=8800),
//          SPACE=(8800,(15,5,2))
//SDSNBKS  DD DSN=&HIQUAL;.SDSNBKS,DISP=(NEW,CATLG),
//          UNIT=&TUNIT;,VOL=SER=&TVOL;;
//          DCB=(RECFM=FB,LRECL=4096,BLKSIZE=28672),
//          SPACE=(28672,(125,5,2))
//SDSNINDX DD DSN=&HIQUAL;.SDSNINDX,DISP=(NEW,CATLG),
//          UNIT=&TUNIT;,VOL=SER=&TVOL;;
//          DCB=(RECFM=FB,LRECL=4096,BLKSIZE=28672),
//          SPACE=(28672,(100,5,1))
//SDSNSHLF DD DSN=&HIQUAL;.SDSNSHLF,DISP=(NEW,CATLG),
//          UNIT=&TUNIT;,VOL=SER=&TVOL;;
//          DCB=(RECFM=VB,LRECL=259,BLKSIZE=8000),
//          SPACE=(8000,(15,5,1))
//*****
//*                DISTRIBUTION LIBRARIES
//*****
//ADSNINST DD DSN=&HIQUAL;.ADSNINST,DISP=(NEW,CATLG),
//          UNIT=&DUNIT;,VOL=SER=&DVOL;;
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=8800),
//          SPACE=(8800,(15,5,2))
//ADSNBKS  DD DSN=&HIQUAL;.ADSNBKS,DISP=(NEW,CATLG),
//          UNIT=&DUNIT;,VOL=SER=&DVOL;;
//          DCB=(RECFM=FB,LRECL=4096,BLKSIZE=28672),
//          SPACE=(28672,(125,5,2))
//ADSNINDX DD DSN=&HIQUAL;.ADSNINDX,DISP=(NEW,CATLG),
//          UNIT=&DUNIT;,VOL=SER=&DVOL;;
//          DCB=(RECFM=FB,LRECL=4096,BLKSIZE=28672),
//          SPACE=(28672,(100,5,1))
//ADSNSHLF DD DSN=&HIQUAL;.ADSNSHLF,DISP=(NEW,CATLG),
//          UNIT=&DUNIT;,VOL=SER=&DVOL;;
//          DCB=(RECFM=VB,LRECL=259,BLKSIZE=8000),
//          SPACE=(8000,(15,5,1))

```

5. Modify DSNDDD for your site, and comment out all DDDEF entries for target and distribution libraries *except* the following:

DDDEF entries for the target zone:

```

REP DDDEF (SDSNINST) DATASET (DSNHELP.SDSNINST) SHR.
REP DDDEF (SDSNBKS)  DATASET (DSNHELP.SDSNBKS)  SHR.
REP DDDEF (SDSNINDX) DATASET (DSNHELP.SDSNINDX) SHR.
REP DDDEF (SDSNSHLF) DATASET (DSNHELP.SDSNSHLF) SHR.
REP DDDEF (ADSNINST) DATASET (DSNHELP.ADSNINST) SHR.
REP DDDEF (ADSNBKS)  DATASET (DSNHELP.ADSNBKS)  SHR.
REP DDDEF (ADSNINDX) DATASET (DSNHELP.ADSNINDX) SHR.
REP DDDEF (ADSNSHLF) DATASET (DSNHELP.ADSNSHLF) SHR.

```

DDDEF entries for the distribution zone:

```

REP DDDEF (ADSNINST) DATASET (DSNHELP.ADSNINST) SHR.
REP DDDEF (ADSNBKS)  DATASET (DSNHELP.ADSNBKS)  SHR.
REP DDDEF (ADSNINDX) DATASET (DSNHELP.ADSNINDX) SHR.
REP DDDEF (ADSNSHLF) DATASET (DSNHELP.ADSNSHLF) SHR.

```

Figure 19. DDDEF entries to include in DSNDDD

Run job DSNDDD before you proceed to the next step.

6. Run the RECEIVE job, DSNREC. This job creates temporary data sets for all online help FMIDs; you cannot receive only the HDB551A FMID.
7. Modify the sample APPLY job, DSNAPPL, to apply only the HDB551A FMID. (Remove JDB5515 and JDB5516 from the APPLY statement.)
8. Modify the sample ACCEPT job, DSNACPT, to accept only the HDB551A FMID. (Remove JDB5515 and JDB5516 from the ACCEPT statement.)
9. Modify sample job DSNUNL2 for your site. This job unloads the books from partitioned to sequential data sets.

Changing Book Data Set Names

Sites have different naming conventions for book data set names. If you change the default book data set names in the sample job, you must also modify the following:

If BookManager READ/MVS is not installed at your site: (you installed the DB2 online help reader):

- Change the QDICTPTH and QLSHELF options in member EOXVOPTS of the SEOYCLIB target library.
- Change the 'Lexis High-Level Qualifier' option in the EOYPARM member of the SEOYCLIB target library
- Change the BKSHELF high-level qualifier in the *your_hlq*.BKLSHELF data set, as shown here:

```
SHELF DSNH0F1 DSNH0F1.BKSHELF DB2 V5 BOOKS FOR ONLINE HELP
```

- Enter the new book data set names on panel DSNTIPA0 when you run the DB2 installation CLIST. Book data set names must end with the suffix .BOOK.

If BookManager READ/MVS is installed at your site:

- Enter the new book data set names on panel DSNTIPA0 when you run the DB2 installation CLIST. Book data set names must end with the suffix .BOOK.

Customizing BookManager READ/MVS

If you are using BookManager READ/MVS (rather than the online help reader), you can customize BookManager for DB2 online help.

- You can add the DB2 library to the BookManager library list. This way, you can select DB2 books from the BookManager library list, as well as from the installation CLIST and DB2I task panels.

To add the DB2 library:

1. Enter BOOKMGR from TSO option 6. BookManager READ logo information is displayed.
2. Press ENTER. A bookshelf list is displayed
3. Select BOOKS on the action bar. The BOOKS pull-down is displayed.
4. Select PERFORM FILE FUNCTIONS. The PERFORM FILE FUNCTIONS window is displayed.
5. Select ADD in the FUNCTION TO PERFORM field. The ADD BOOKSHELF window is displayed.

6. Type the DB2 bookshelf data set name (default= 'DSNSH0F1.BKSHELF') in the DATA SET NAME field and press ENTER. The BOOKSHELF LIST DATA SET TO BE MODIFIED window is displayed. The DATA SET NAME field contains the name of your personal bookshelf list.
7. Replace the bookshelf list data set name in the DATA SET NAME field with the data set name of your site's system bookshelf list. This data set name is specified in option QLSHELF of the EOXVOPTS member in the SEOYCLIB target library.

The BOOKSHELF LIST DATA SET TO BE MODIFIED window is removed and the bookshelf is now in the list.

8. Press **PF5** to refresh the bookshelf list. The DB2 bookshelf is now on the list.
- You can suppress the BookManager logo information so that the logo does not display each time you enter online help. To suppress the logo information, modify the EOXVSTRT member of the SEOYCLIB target library by adding the following line after the trace command at the beginning of the file:

```
ISPEXEC CONTROL NONDISPL ENTER /* Suppress logo information */
```

- You can build a searchable bookshelf. The bookshelf shipped on the tape is not enabled for cross-book searches. To build a searchable bookshelf:
 1. From the bookshelf list, open the DSNSH0F1 bookshelf. A list of DB2 books is displayed.
 2. Select SEARCH on the action bar. The SEARCH menu is displayed.
 3. Select option 2 'Set up search ...'. The following error message is displayed:

```
Books View Search Group Options Help
```

```
-----  
Severe Read or Search Error
```

```
A serious BookManager error has occurred while reading or searching the  
current bookshelf or book. Processing for the current bookshelf or book  
is terminated.
```

```
Reason . . : Bookshelf does not match bookshelf search index  
Data set . . : 'DSNHELP.DSNSH0F1.BKSHELF'  
Routine . . : EOXMPLSH:plsh_get_string  
Code . . . : 9
```

```
F1=Help F12=Cancel  
-----
```

4. Press ENTER. Choose option 1 on the following dialog:

```
Books View Search Group Options Help
```

```
-----  
Rebuild Bookshelf
```

```
The bookshelf does not match its bookshelf search index. Select  
one of the following actions, and then press ENTER.
```

- ```
1 1. Rebuild the bookshelf
 2. Do not rebuild the bookshelf
```

```
| Data set name . : 'DSNHELP.DSNH0F1.BKSHELF'
```

```
F1=Help F12=Cancel
```

5. The following message appears:

```
| The bookshelf data set 'DSNHELP.DSNH0F1.BKSHELF' has been updated to
 correspond with its search index. You must refresh the bookshelf to use it.
```

6. Exit the bookshelf and select F5 to refresh.

---

## Verifying Online Help and Setting Exit Options

Before installing DB2, you must verify that DB2 Online Help is set up correctly. You can also turn off the confirming messages that appear when you exit an online book.

**To verify that online help is installed correctly:** From the TSO command option, enter:

```
BOOKMGR
```

```

#
```

If you are using the Online Help Reader, do not use the Help pulldown menu from the BookShelf list.

The bookshelf list should be displayed, as in this example:

```

Books View Options Help

Command ==> _____ SCROLL ==>

 Bookshelf List
 Shelves 1 to 1

Shelf Name Description
__ DSNSH0F1 DB2 V5 BOOKS FOR ONLINE HELP

```

If the BOOKMGR command abends, verify that you have concatenated the BookManager data sets correctly in the LINKLIST, in the TSO logon procedure, and in any user-supplied ISPF start-up CLISTS.

If the BOOKMGR command results in a "data set not found" message, verify that you have completed the tasks in "Unloading DB2 Books" on page 2-59.

**To set the exit options:**

1. Select **Options** on the action bar. The **Options** pull-down is displayed.
2. Select **Set exit options**. The **Set Exit Options** window is displayed.
3. Turn off the exit confirmations and bookmarks, as in this example:

```

 Set Exit Options

Exit from book 2 1. Confirm exit
 2. Do not confirm exit

Setting of bookmark . . . 3 1. Keep current closing bookmark
 2. Place the closing bookmark
 3. Exit without closing bookmark

Exit from bookshelf and
bookshelf list 2 1. Confirm exit
 2. Do not confirm exit

Save the changes as . . . 1 1. Permanent
 2. Temporary

```

4. Press ENTER to save the changes.
5. Press the exit key (**F3**) to exit online help.



---

## Using Online Help

DB2 Online Help links you directly from DB2 panels to the information you need in the DB2 online books.

## Accessing Help with DB2 Online Help

To get help for a DB2 installation or DB2I panel, press the Help PF key (usually **F1**). A selection panel for help topics appears. Type the number of the topic you want help on and press Enter. The online help reader starts and opens a DB2 online book to the topic.

## Moving Around in the DB2 Online Help

The online help reader allows you to move around the online book in several ways:

- Press **F8** to go forward one panel and **F7** to go backward one panel.
- Text containing hypertext links is highlighted; move the cursor to the highlighted area and press Enter to link to related information on the topic.
- Graphics have a PICTURE label with a number. Move the cursor to the label and press Enter to display the graphic. Press **F3** to return to the text. If you do not have GDDM installed, you cannot view graphics.
- Figures and tables may be wider than the display screen. Type RIGHT on the command line to scroll to the right of the figure or table; type LEFT to return to the left margin.

See “Searching for Additional Information” for additional navigation information.

### Searching for Additional Information

To search the book you are currently viewing, select **Search** on the action bar and then select **Set up search** in the **Search** pull-down. Type the information you want to find in the **Search for** area of the **Set up search** window. Your search request can be up to 44 characters long, and can include any combination of words, phrases, and special characters.

**Selecting the Search Match Type:** The default search match type is *fuzzy matching*. With fuzzy matching, the online help reader looks for words that share the same language root as words in your search request, such as the plural forms of nouns or different tenses of verbs. You can specify *exact matching, any case* when you want to find the exact words you type in the **Set up search** window regardless of capitalization, or *exact matching, including case* to find the exact words including capitalization and punctuation. To change the search match type:

1. Select **Search** on the action bar
2. Select **Set up search** in the **Search** pull-down
3. Press **F2** in the **Set up search** window
4. Move the cursor to the type of search matching you want to use
5. Either:
  - Press Enter to change the search match type temporarily, or
  - Press **F2** to set the search match type as your new default

**Tailoring Your Search Request:** To tailor your search request, you can:

- Include a space between words, as in white house. A topic matches if it contains both words separated by any number of spaces in the text, but not by punctuation.
- Use a phrase separator. Type the separator character (the default is a comma) after each word or phrase you want to separate. A topic matches if it contains either word.
- Use a pattern-matching character. Type the pattern character (the default is an asterisk) in place of characters at the end of the word, as in hous\*. Words that start with the specified characters match.

**Searching for Variations and Synonyms:** You can also search for variations and synonyms of words. To search for spelling variations:

1. Type your search request in the **Set up search** window
2. Move the cursor to a word in the request and press **F4**
3. You see a list of spelling variations for that word in the **Wordcheck** window.
  - To add words to your search request, type a slash (/) next to each word you want to include and press **F4**.
  - To replace a word in your search request, type a slash (/) next to the word you want to use and press **F5**
4. Press Enter to search the book

To search for synonyms:

1. Type your search request in the **Set up search** window
2. Move the cursor to a word in the request and press **F5**
3. You see a list of synonyms for that word in the **Synonyms** window.
  - To add words to your search request, type a slash (/) next to each word you want to include and press **F4**.
  - To replace a word in your search request, type a slash (/) next to the word you want to use and press **F5**
4. Press Enter to search the book

**Working with Your Search Matches:** When you search a book, DB2 Online Help displays a list of the topics containing information that matches your search request. The online help reader displays the topics in the list by location, frequency and size, exactness, uniqueness, and similarity. The topics that appear first on the list are those with the highest ranking.

- To see the context of the match, press **F4** to see a line of text beneath each topic entry that shows where the best search match in the topic occurred.
- To see an explanation for why a topic matches your search request, move the cursor to the topic identifier and press **F6**.
- To view a topic with a search match, move the cursor to the topic identifier and press Enter. You go to the first occurrence of the search word or phrase in the selected topic.
- To bring up your search list again, select **List all topics with matches** from the **Search** pull-down.

Other options from the **Search** pull-down include **Go to next match**, **Go to next best topic**, and **Emphasize matches** (to highlight matching text in the book).

### **Exiting DB2 Online Help**

To exit the online help reader, press the Exit PF key (usually **F3**). The selection panel for the DB2 task panel you were working with appears. To exit DB2 Online Help and return to the task panel, press the Exit key again.

---

## **Maintaining Online Help**

Updated versions of DB2 books are available on the DB2 for OS/390 Version 5 Licensed Online Library CD-ROM. If your site makes use of CD-ROM updates, you can make the update books accessible to the online help reader as follows:

- For the installation CLIST:

Enter the new book data set names on panel DSNTIPA0. Book data set names must end with the suffix `.BOOK`.

- For DB2I:

Select Option 10 on the DB2I Defaults Panel and enter the new book data set names. You must have write access to the CLIST specified on the DB2I panel DSNEIPA0.



---

## Chapter 2-4. Loading DB2 Libraries

IBM distributes DB2 on tapes or cartridges, depending on which feature you order.

**If you are installing DB2**, your first task is to load the data sets on the distribution tapes or cartridges into DB2 libraries.

**If you are migrating** to Version 5, you need to check *DB2 Program Directory* to ensure that you are at the proper maintenance level *before* you load the data sets on these tapes or cartridges into DB2 libraries.

To load the DB2 libraries, use System Modification Program Extended (SMP/E). SMP/E processes the installation tapes or cartridges and creates DB2 distribution libraries, DB2 target libraries, and SMP/E control data sets.

DB2 provides several jobs that invoke SMP/E. These jobs are distributed on one of the tapes or cartridges you receive.

# If you ordered a CBIPO, CBPDO, or ServerPac refer to the documentation sent  
# with that package for instructions on loading your DB2 libraries. Continue your DB2  
# installation according to the instructions in "Chapter 2-5. Installing, Migrating, and  
# Updating System Parameters" on page 2-89.

---

### What IBM Sends You

| When you order DB2, you receive six standard label 9-track magnetic tapes  
| recorded at 6250 BPI, six 3480 cartridges, or six 4mm cartridges, depending on the  
| feature you ordered. If you order a custom built product delivery offering (CBPDO),  
| your order may differ.

# DB2 Installer, Estimator, Visual Explain, DB2 Connect, DB2 for OS/390 Control  
# Center enablement, and DB2 REXX Language Support are separately ordered as  
# non-priced features of DB2.

# DB2 Installer, Estimator, Visual Explain, and DB2 Connect are workstation features  
# that are shipped on CD ROMs. For more information on these features see their  
# readme files on the CDs. The DB2 for OS/390 Control Center enablement is  
# shipped on a tape or cartridge. Directions for installing the control center  
# enablement are in the DB2 Management Tools Package Program Directory.

# DB2 REXX Language Support lets you write and run REXX language applications  
# that include SQL statements. For more information on installing DB2 REXX  
# Language Support, see this chapter and the DB2 REXX Language Support  
# Program Directory. For information on using DB2 REXX Language Support, see  
# *Application Programming and SQL Guide*.

The tapes or cartridges are in SMP RELFILE format. The first file of each of these tapes or cartridges contains SMP/E modification control statements in RELFILE format. All succeeding files contain IEBCOPY unloaded partitioned data sets for SMP/E to process.

Each tape or cartridge has one or more function modification identifiers (FMIDs) that SMP/E uses to distinguish separate parts of DB2. This arrangement simplifies

shipping and service. IRLM, for example, is distributed with both IMS and DB2, and therefore has a separate FMID. Even though your site might not use every module, you must load each FMID.

Along with these tapes or cartridges, you receive a set of documents. One of these documents is *IBM DATABASE 2 Program Directory*. Read the Program Directory before installing or migrating DB2. It identifies and describes the contents of FMIDs for each tape or cartridge. It also describes any additional service that needs to be applied to DB2. You also receive DATABASE 2 Performance Monitor's program directory. Read this directory before installing any of the DB2 PM features.

If you plan to use DB2's callable SQL interface (DB2 Call Level Interface), there are additional installation jobs that you need to run. See *Call Level Interface Guide and Reference* for more information.

If you plan to use DB2 for OS/390 Java™ Edition, see *Application Programming Guide and Reference for Java* for additional installation jobs that you need to run.

Before installing DB2, use Information/Access or the ServiceLink facility of IBMLink to check for PSP updates to the information contained in both the Program Directory and this book. Refer to *DB2 Program Directory* for PSP keyword specifications. Be sure that you apply all necessary corrective service to your DB2 system before migrating. It is also a good idea to check monthly for PSP updates. This way, you get the most current information about DB2. Contact the IBM Support Center if you do not have access to IBMLink.

---

## What You Produce

During SMP/E processing, DB2 loads the distribution and target libraries. The distribution libraries are used to maintain DB2 and contain the master copy of all elements for your DB2 system. The target libraries contain the various DB2 components. DB2 target libraries are updated by corrective service.

Table 38 and Table 39 on page 2-71 describe all the DB2 distribution and target libraries.

The storage requirements for target and distribution libraries are listed in "DASD Requirements for the DB2 Libraries and SMP/E Data Sets" on page 2-16.

Table 38. DB2 Distribution Libraries

| Distribution Libraries            | Description                                                                             |
|-----------------------------------|-----------------------------------------------------------------------------------------|
| <i>prefix</i> .ADSNLOAD           | The distribution libraries contain the master copy of all elements for your DB2 system. |
| <i>prefix</i> .ADSNMACS           |                                                                                         |
| <i>prefix</i> .ADSNENU or ADSNDKF |                                                                                         |
| <i>prefix</i> .ADXRLoad           |                                                                                         |
| <i>prefix</i> .ADXRSAMP           |                                                                                         |
| <i>prefix</i> .ADSNDKF            |                                                                                         |

Table 39. DB2 Target Libraries

| Target Libraries                                   | Description                                                                                                                                                                                                                                                  |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>prefix</i> .SDSNCLST                            | This TSO CLIST library contains code used to simplify the process of installing and migrating, to aid program preparation and the use of DB2 utilities, and to allow the use of DB2 Interactive (DB2I).                                                      |
| <i>prefix</i> .SDSNDBRM                            | This library contains the system DBRMs for DB2 Version 5.                                                                                                                                                                                                    |
| <i>prefix</i> .SDSNEXIT                            | This program library is empty when first created. The installation jobs put members into this library.                                                                                                                                                       |
| <i>prefix</i> .SDSNLINK                            | This library contains ERLY code of Version 5.                                                                                                                                                                                                                |
| <i>prefix</i> .SDSNLOAD                            | This library contains Version 5 load modules.                                                                                                                                                                                                                |
| <i>prefix</i> .SDSNMACS                            | This macro library contains macros needed for the CICS and IMS attachment facilities, the initialization parameter macros, and some data mapping macros needed for some applications.                                                                        |
| <i>prefix</i> .SDSNSAMP                            | This initialization library contains the sample applications and data, the jobs for installing and migrating, the default installation and migration parameters, and catalog initialization data for DB2. The JCLIN for each FMID is stored in this library. |
| <i>prefix</i> .SDSNSPFM                            | This DB2 ISPF message library contains messages issued during install or migrate processing.                                                                                                                                                                 |
| <i>prefix</i> .SDSNSPFP                            | This is the DB2 ISPF library for installation task and help panels.                                                                                                                                                                                          |
| <i>prefix</i> .SDSNSPFS                            | This is the DB2 ISPF skeleton library used to produce EDITJCL.                                                                                                                                                                                               |
| <i>prefix</i> .SDSNSPFT                            | This is the DB2 ISPF command table library.                                                                                                                                                                                                                  |
| <i>prefix</i> .SDSNPFPE or <i>prefix</i> .SDSNPFPK | <i>prefix</i> .SDSNPFPE contains the English task and help panels, and <i>prefix</i> .SDSNPFPK contains the Kanji task and help panels.                                                                                                                      |
| <i>prefix</i> .SDSNENU or <i>prefix</i> .SDSNDKF   | <i>prefix</i> .SDSNENU contains the installation procedures for installing the English task and help panels, and <i>prefix</i> .SDSNDKF contains the installation procedures for installing the Kanji task and help panels.                                  |
| <i>prefix</i> .SDSNC.H                             | This contains the CLI header files.                                                                                                                                                                                                                          |
| <i>prefix</i> .SDXRSAMP                            | IRLM samples library.                                                                                                                                                                                                                                        |
| <i>prefix</i> .SDXRRESL                            | This library contains the IRLM load modules.                                                                                                                                                                                                                 |

The remainder of this chapter explains how to edit and run all the SMP/E jobs that DB2 provides. These jobs allocate the DB2 libraries and load them with the data from the installation tapes or cartridges.

For a description of each job, see Table 40.

Table 40 (Page 1 of 2). List of SMP/E Jobs

| Job Name | Description                                                                                  |
|----------|----------------------------------------------------------------------------------------------|
| DSNTIJAC | These jobs invoke SMP/E to accept all the FMIDs into the DB2 distribution libraries (DLIBs). |

Table 40 (Page 2 of 2). List of SMP/E Jobs

| Job Name | Description                                                                                                                                           |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| DSNTIJAE | This is the SMP/E <i>allocation job</i> . It creates and catalogs the SMP/E control data sets and the DB2 target and distribution libraries.          |
| DSNTIJAP | This job invokes SMP/E to apply all the FMIDs to the DB2 target libraries.                                                                            |
| DSNTIJRC | This job invokes SMP/E to receive all the FMIDs (from both tapes or cartridges) into the SMP/E control data sets.                                     |
| DSNTEJAC | This job invokes SMP/E to accept the FMIDs for the English DB2I panels into the DB2 distribution libraries (DLIBs).                                   |
| DSNTEJAE | This SMP/E allocation job creates and catalogs the SMP/E control data sets and the DB2 target and distribution libraries for the English DB2I panels. |
| DSNTEJAP | This job invokes SMP/E to apply all the FMIDs for the English DB2I panels to the DB2 target libraries.                                                |
| DSNTEJRC | This job invokes SMP/E to receive all the FMIDs (from both tapes or cartridges) for the English DB2I panels into the SMP/E control data sets.         |
| DSNTNJAC | These jobs invoke SMP/E to accept all the FMIDs for the Kanji DB2I panels into the DB2 distribution libraries (DLIBs).                                |
| DSNTNJAE | This SMP/E allocation job creates and catalogs the SMP/E control data sets and the DB2 target and distribution libraries for the Kanji DB2I panels.   |
| DSNTNJAP | This job invokes SMP/E to apply all the FMIDs for the Kanji DB2I panels to the DB2 target libraries.                                                  |
| DSNTNJRC | This job invokes SMP/E to receive all the FMIDs (from both tapes or cartridges) for the Kanji DB2I panels into the SMP/E control data sets.           |
| DSNTTJAC | These jobs invoke SMP/E to accept all the FMIDs for DB2 REXX Language Support into the DB2 distribution libraries (DLIBs).                            |
| DSNTTJAP | This job invokes SMP/E to apply all the FMIDs for DB2 REXX Language Support to the DB2 target libraries.                                              |
| DSNTTJRC | This job invokes SMP/E to receive all the FMIDs (from both tapes or cartridges) for DB2 REXX Language Support into the SMP/E control data sets.       |
| DSNTIJUD | Job DSNTIJUD invokes SMP/E to delete all DB2 for MVS/ESA Version 4 entries from the SMP/E libraries.                                                  |

#  
#  
#  
#  
#  
#  
#  
#  
#

## SMP/E Step 1: Copy and Edit the SMP/E Jobs

Before running any of the SMP/E jobs, you must copy them from the tape or cartridge on which they are distributed to a disk that you define. To do this, use the sample JCL that appears in Figure 20 on page 2-73.

If you have a CBIPO or CBPDO, refer to the documentation sent with the package.



## Copying the SMP/E Jobs

This JCL invokes the MVS utility IEBCOPY to copy the jobs to DASD. It then invokes the MVS utility IEBTPCH to print each job. If you need additional information about these utilities, see *DFSMS/MVS: Utilities*.

```
//* COMPID: DB2,5740XYR00
//* DOC: LOAD SMP INSTALLATION JCL FROM TAPE FOR DB2
//LOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//JCLTAPE DD DSN=IBM.HDB5510.F2,DISP=(OLD,PASS),
// UNIT=TAPE,VOL=(PRIVATE,,SER=DB5510),
// LABEL=(3,SL)
//*
//JCLDISK DD DSN=SYSADM.JCL.CNTL,DISP=OLD
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
COPY I=JCLTAPE,O=JCLDISK
SELECT MEMBER=(DSNTIJAE)
SELECT MEMBER=(DSNTIJAC,DSNTIJAP,DSNTIJRC,DSNTIJUD)
//*
//PRINT EXEC PGM=IEBTPCH
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=SYSADM.JCL.CNTL,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD *
PRINT TYPORG=PO,MAXFLDS=5
RECORD FIELD=(80)
//*
```

Figure 20. Sample JCL to Copy SMP/E Jobs to DASD

If you are using a 3480 cartridge, remove the DCB parameter and change UNIT=TAPE to the appropriate device type. Also, the job assumes that the data set SYSADM.JCL.CNTL already exists. Either allocate it or use another data set. Finally, tailor the JCL to reflect the unit names and volume serial numbers that your site uses.

If this job fails or abends, correct the job and rerun it.

This JCL copies and prints six members. These jobs currently exist as members of the partitioned data set IBM.HDB5510.F2 on tape VOLSER=DB5510.

After running the copy job, edit and run jobs DSNTIJAE, DSNTIJUD (optional), DSNTIJRC, DSNTIJAP, and DSNTIJAC. See the header notes within each job for information on how to customize the job for your particular installation.

## Editing the SMP/E Jobs

Before running any of the SMP/E jobs, you must edit them. This section identifies the items you might want to modify. The chart below lists each of these items and the page on which a description appears. Read the entire section before you begin making changes.

| Item           | Page |
|----------------|------|
| JOB Statements | 2-74 |

| Item                              | Page |
|-----------------------------------|------|
| Link List Options                 | 2-74 |
| DB2 Program Library               | 2-76 |
| DB2 Library Naming Considerations | 2-78 |
| SMP/E Data Set Options            | 2-79 |
| IRLM Options                      | 2-80 |

## Creating JOB Statements

The SMP/E jobs do not include JOB statements. Although JOB statements are often built automatically, it is usually easier for you to create JOB statements that are correct for your site than to edit provided JOB statements. You can do one of the following:

- If you are using ISPF to edit and submit the SMP/E jobs, edit a member containing the JOB statement. Delete all text except the job statement. Then use the ISPF COPY command to copy the member into each job before submitting it.
- If you are using TSO to submit the SMP/E jobs, edit a JOB statement and submit that JOB statement with each job.

For example, data set JCL.CNTL(J) might contain the following:

```
//DB2INST JOB ACCT,NAME,
// MSGCLASS=A,MSGLEVEL=(1,1),
// TIME=(10),USER=SYSADM,PASSWORD=xxxxxxxx
/*JOBPARM
/*ROUTE PRINT
```

When you are ready to submit a job, use a command like the following:

```
SUBMIT (JCL(J) JCL(DSNTIJxx))
```

where *xx* are the last two characters of the SMP/E job name. This command submits the JOB statement along with the job.

## Choosing Link List Options

Link list options for the three load module libraries are as follows:

*prefix.SDSNLINK*: Contains modules that you must place in the link list because they are loaded at subsystem initialization during IPL. For Version 5, the load module library SDSNLINK contains modules that are called early (ERLY) code. If your system is at the prerequisite maintenance level, your DB2 for MVS/ESA Version 4 ERLY code is *upward* compatible with DB2 for OS/390 Version 5. The Version 5 early code is *downward* compatible with Version 4.

Be aware that some early code modules contain SQLSTATEs. In Version 4, some SQLSTATE values were changed or added to conform to ANSI/ISO SQL standard of 1992. You might need to modify any application programs that reference SQLSTATE values from a previous release. See *Messages and Codes* for a list of SQL codes that are changed or added. As mentioned earlier, it is a good idea to check monthly for PSP updates to get the most current information about DB2.

**If you are migrating**, be aware that any maintenance to ERLY code or installation of new ERLY code requires that you IPL MVS to execute the ERLY code. Pointing

to SDSNLINK, STEPLIB, LLA REFRESH, or stopping LLA **fails** to update the MVS Subsystem Vector Table (SSVT). See *DB2 Program Directory* for details.

Schedule an MVS IPL before or during a migration to a new release of DB2. This is necessary because migration job DSNTIJMV makes changes to SYS1.PARMLIB that are not recognized by MVS until the next IPL. Changes that DSNTIJMV makes to the SYS1.PARMLIB affect the following:

- New subsystem definitions in IEFSSNxx
- New APF libraries in IEAAPFxx
- New load module libraries in LNKLSTxx.

***prefix.SDSNLINK:***

- Contains early code
- Is shareable by multiple subsystems and releases of DB2
- Is APF-authorized

***prefix.SDSNLOAD:***

- Contains modules that you can place in the MVS link list
- Is a main load module repository
- Is shareable by multiple subsystems at same release level
- Allows only DB2 to modify code
- Holds default exits
- Is APF-authorized

***prefix.SDSNEXIT:***

- Contains modules that you can place in the link list
- Holds the subsystem parameter module, DSNHDECP, and user-written exits
- Is modified by user
- Is APF-authorized

Libraries *prefix.SDSNLOAD* and *prefix.SDSNEXIT* are separate to allow users who are supporting two levels of DB2 to access modules from either level by using STEPLIB and JOBLIB statements. This also minimizes the number of IPLs required by corrective service to DB2 load modules, and it reduces the size of the LNKLST lookaside (LLA) list. When *prefix.SDSNLOAD* and *prefix.SDSNEXIT* are used together, list *prefix.SDSNEXIT* first to override the IBM defaults in *prefix.SDSNLOAD*.

# ***IRLM link list requirement:*** You must add the IRLM load module DXRRL183 to the link list. This requires that you copy the module into another library. After you apply maintenance to IRLM that affects DXRRL183, remember to copy the updated module to the link list.

***Supporting One DB2 Subsystem:*** There are several methods of maintaining a single DB2 subsystem. The following steps describe what is probably the easiest method for most sites:

1. Change the SMP/E procedure DSNTIJAE to assign all load modules to *prefix.SDSNLOAD*. You can do this by changing the data set name for DDDEF (SDSNLINK) from *prefix.SDSNLINK* to *prefix.SDSNLOAD*.
2. Remove the allocation for *prefix.SDSNLINK* from the allocation job DSNTIJAE.

3. Include *prefix*.SDSNLOAD (instead of *prefix*.SDSNLINK) in the LNKSTxx member of SYS1.PARMLIB.

**Supporting Multiple DB2 Subsystems:** Supporting multiple subsystems can mean several things. You can have two or more DB2 subsystems at the same release and service level (for instance, two DB2 Version 5 subsystems). If this is the case, read the suggestions on page 2-75. In addition, create separate libraries for DSNHDECP and user-written exits of each DB2 subsystem. For considerations in data sharing environments, see *Data Sharing: Planning and Administration*.

You can also have two or more DB2 subsystems at the same release level, but at different service levels. For instance, you can have a DB2 Version 5 production subsystem and a DB2 Version 5 test subsystem at different service levels. Or, you can have two DB2 subsystems at different release levels. For instance, you can have a DB2 for OS/390 Version 5 subsystem and a DB2 for MVS/ESA Version 4 subsystem.

In either of these cases, you can assign the DB2 modules that must be in the link list libraries to an existing link list data set. To do this, change the data set name for DDDEF (SDSNLINK) in the DSNTIJAE procedure to the name of an existing entry in the LNKSTxx member of SYS1.PARMLIB. You might still want to have the *prefix*.SDSNLOAD data set listed once in the link list to permit fewer STEPLIB statements. With different SDSNEXIT data sets, you can easily have different subsystem parameter or DSNHDECP members for each subsystem.

### Accessing the Correct DB2 Program Library

If you do not place *prefix*.SDSNLOAD in the LNKSTxx member of SYS1.PARMLIB, you must provide JOBLIB or STEPLIB statements for it in certain types of programs and procedures.

The installation and migration jobs provided with DB2 Version 5 already contain the necessary JOBLIB or STEPLIB statements. In addition, the startup procedures that DB2 provides for DB2 for MVS/ESA Version 4 and DB2 for OS/390 Version 5 include STEPLIB statements for their respective program libraries, *prefix*.SDSNLOAD and *prefix*.SDSNLOAD.

Provide STEPLIB or JOBLIB statements for the following types of programs and procedures if you do not place *prefix*.SDSNLOAD in the LNKSTxx member of SYS1.PARMLIB.

- **TSO or batch jobs** that access DB2 services require JOBLIB or STEPLIB statements for *prefix*.SDSNLOAD. These jobs include TSO logon procedures and batch jobs that access the DSN command and subcommands, the DB2 precompiler, and DB2 utilities.
- **IMS control, message, and batch processing jobs** also require JOBLIB or STEPLIB statements for *prefix*.SDSNLOAD. You must specify the DB2 load library in the startup procedure for each IMS region (IMS control, message processing program (MPP), batch message processing (BMP), and Fast Path region) that can communicate with DB2. You can do this in two ways:
  1. If all the data sets referred to in the JOBLIB or STEPLIB statement for an IMS region are APF-authorized, then add the DD statement for *prefix*.SDSNLOAD to the JOBLIB or STEPLIB statement. If you are using the DYNAM option of COBOL II, the IMS RESLIB DD statement must

precede the reference to *prefix*.SDSNLOAD in the JOBLIB or STEPLIB statement.

2. If any of the data sets referred to in the JOBLIB or STEPLIB statement for the IMS region are not APF-authorized, then add the DFSESL DD statement for *prefix*.SDSNLOAD. All libraries specified on the DFSESL DD statement must be APF-authorized. The DFSESL DD statement is not required by the DB2 DL/I Batch support. IMS requires that an IMS RESLIB DD statement also be referred to by the DFSESL DD statement, as in the following:

```
//DFSESL DD DSN=ims_reslib,DISP=SHR
// DD DSN=prefix.SDSNLOAD,DISP=SHR
```

- **CICS procedures**, including the CICS initialization JCL, also need to include DB2 libraries. See “Updating CICS Initialization JCL” on page 2-396 for more information.

The migration jobs include a step to rename old procedures before adding new ones. Before renaming the jobs, check existing ones so you do not overwrite them. After renaming the jobs, update the procedures to include STEPLIB or JOBLIB statements to use the appropriate load module libraries.

## Performance Considerations

This section discusses performance considerations for including modules in the libraries that are included in the link list and presents some suggestions on the strategies you might want to pursue. These general suggestions might not match the specific needs of your site.

Adding many modules to the libraries included in the link list can reduce system performance. However, adding only a few modules to the libraries requires additional STEPLIB or JOBLIB statements. Because these STEPLIB or JOBLIB statements must be searched before the link list is searched, this approach can also reduce system performance. The approach that produces the best performance for your site depends on the environment in which you use DB2. Regardless of which attachment facilities you use, the modules in *prefix*.SDSNLINK must always be in the link library list.

If you are using DB2 with IMS, you probably want to include *prefix*.SDSNLINK, not *prefix*.SDSNLOAD, in the LNKSTxx member of SYS1.PARMLIB, because both the IMS RESLIB and *prefix*.SDSNLOAD have the DSNHLI alias. Place the needed STEPLIB or JOBLIB statements in the IMS procedures.

If you are using DB2 with IMS and you want *prefix*.SDSNLOAD (in addition to *prefix*.SDSNLINK) in the LNKSTxx member of SYS1.PARMLIB, be sure that the library concatenation for *prefix*.SDSNLOAD and the IMS RESLIB are correct for your site, because both libraries have the DSNHLI alias.

If you are using DB2 with CICS, you probably want to put *prefix*.SDSNLINK, not *prefix*.SDSNLOAD, in the LNKSTxx member of SYS1.PARMLIB. Then place the needed STEPLIB or JOBLIB statements in the CICS procedures.

The approach for using the TSO and call attachment facilities involves the following considerations:

- If you use the DSN command and its subcommands infrequently, place *only prefix*.SDSNLINK in the LNKSTxx member of SYS1.PARMLIB. Provide the

necessary STEPLIB or JOBLIB statements in your TSO logon procedures or in your JCL if you are using batch.

- If you use the DSN command and its subcommands frequently, you might also want to move the TSO attach load modules to a library defined in the LNKLSTxx. The TSO attach modules are DSNECP00, DSNECP10, DSNESM00, and DSNELI.
- If you use the call attachment facility (CAF) frequently, move the CAF load modules (DSNACAB, DSNACAF, and DSNALI) to a library defined in the LNKLSTxx.
- If you use the CAF or the DSN command and its subcommands frequently, you might also want to move the eligible load modules to a library defined in the link pack area (LPA), IEALPAXx member of SYS1.PARMLIB. The CAF and DSN load modules must reside below the 16MB line of MVS virtual storage.
  - The TSO load modules that you can place in the LPA are DSNECP00, DSNECP10, DSNESM00, and DSNELI. If you include these modules in the LPA, do not forget to include the appropriate aliases for DSNECP00 (DSN) and DSNELI (DSNHLI).
  - The CAF load modules that you can place in the LPA are DSNACAF and DSNALI. If you include these modules in the LPA, do not forget to include the appropriate alias for DSNALI (DSNHLI2). Do not include DSNACAB in the LPA because it is a data-area-only, non-executable load module.

**Attention:**

If modules are moved or copied from one library to another, changes must be made to SMP/E control data to reflect the movement. If you do not make these changes, future service or changes to the modules will not be processed correctly.

**DB2 Library Naming Considerations**

You need to modify the DB2 library data set names in the SMP/E jobs. These data sets are listed in Table 38 on page 2-70. Their names are composed of three parts:

- A user-defined prefix
- A fixed base name: for example, SDSNLOAD
- An optional user-defined suffix

The Version 5 default prefix *prefix* is used in this book; the default suffix is null. You need to edit each of the DB2 SMP/E jobs and follow the directions in the header notes of each job to specify the names of the SMP/E data sets. If you want to add a suffix, edit the SMP/E procedures and allocation jobs. The prefix cannot exceed 18 characters. The suffix cannot exceed 17 characters, minus the length of the prefix. In addition, any data set names exceeding eight characters must be in groups of no more than eight characters, separated by periods. The qualified data set name cannot exceed 44 characters.

You can also change the base name of these libraries or load them into another data set. If you do this, however, you might need to do additional editing of the installation or migration jobs. The DSNTINST CLIST, which you use later to tailor the installation and migration jobs, uses the following default data set names:

*prefix*.ADSNLOAD

*prefix*.SDSNMACS

#

*prefix*.SDSNCLST  
*prefix*.SDSNEXIT  
*prefix*.SDSNLINK  
*prefix*.SDSNLOAD  
*prefix*.SDSNDBRM

*prefix*.SDSNSAMP  
*prefix*.DBRMLIB.DATA  
*prefix*.RUNLIB.LOAD  
*prefix*.SRCLIB.DATA  
*prefix*.SDXRRESL

We recommend that you use the supplied naming convention.

Document any changes you make to the library names in the SMP/E jobs. You must specify these library names again during the ISPF tailoring session.

## **SMP/E Data Set Options**

You have several options regarding how you establish and use SMP/E data sets. You must decide whether you will have DB2 and IMS share SMP/E data sets. You must also decide whether you need an additional set of SMP/E data sets. An additional set of SMP/E data sets is required if you are supporting more than one release of DB2.

**Sharing SMP/E Data Sets with IMS:** If you do not share SMP/E data sets with IMS, skip this section and continue reading with 'Establishing SMP/E Data Sets for Two Releases' on page 2-80.

DB2 and MVS cannot share SMP/E data sets because there are module names and macro names common to both products. Under certain conditions, however, DB2 can share SMP/E data sets with IMS.

The allocation job you run, DSNTIJAE, defines a new set of SMP/E data sets that DB2 and IMS will share.

**Sharing SMP/E Data Sets with CICS:** The CICS - DB2 attachment facility feature (JC14106) on the CICS Version 4 product tape contains some macros with the same name as macros on the DB2 tape. To prevent existing modules with the same names from being overwritten, do not install CICS/ESA Version 4 into the same target and distribution zones as DB2.

You must modify your allocation job for either of the following situations:

**Situation 1:** You need to or want to have separate SMP/E data sets for DB2 and IMS. In certain instances, DB2 and IMS cannot share SMP/E data sets. If your version of IMS is not Version 2.2 or later, you *must* have separate SMP/E data sets for DB2 and IMS. You also must have separate SMP/E data sets to have two IRLMs.

Even if you are not *required* to have separate SMP/E data sets, you might want them separate anyway. If DB2 and IMS share the SMP/E data sets, you need to accept or reapply DB2 corrective service to these data sets to allow IMS SYSGENs.

To establish separate SMP/E data sets for DB2 and IMS, change the data set prefix that your allocation job uses to a value other than the prefix you use for your current IMS SMP/E data sets. The allocation jobs use the prefix *IMS*. Changing this prefix prevents the allocation job from replacing your current SMP/E data sets and still allows it to create new SMP/E data sets.

**Situation 2:** You want to share SMP/E data sets between DB2 and IMS, but you want to use the SMP/E data sets that already exist for IMS. To do this, remove the data set allocation and initialization statements from your allocation job. When you run the job, no SMP/E data sets will be created, and DB2 will share the existing SMP/E data sets with IMS.

For additional information about sharing data sets, refer to *System Modification Program Extended (SMP/E) User's Guide*.

**Establishing SMP/E Data Sets for Two Releases:** A single set of SMP/E zone structure can record only one release of DB2. We strongly recommend that you maintain separate zone structures for both DB2 for MVS/ESA Version 4 and DB2 for OS/390 Version 5 until you are sure that you will not fall back. The SMP/E jobs provided with DB2 assume that you will allocate a new set of SMP/E data sets for the new release. When you run your allocation job (DSNTIJAE), it creates a set of SMP/E data sets. If you choose to reuse your Version 4 zone structure, you can run job DSNTIJUD to delete SMP/E data for Version 4. However, after you run this job, you cannot fall back.

You can create an additional set of SMP/E data sets either by copying them from a prior release of DB2 or by allocating a new set. Allocating a new set is faster because no data must be deleted. However, we recommend copying a prior set because it allows you to perform service regression checking.

### IRLM Options

The SMP/E prefix in the SMP/E jobs is the same for the new IRLM as for the old IRLM. Consequently, if you do not change the SMP/E prefix, the jobs will overwrite your old IRLM. If you do not want to do this, edit the jobs accordingly.

---

## SMP/E Step 2: Allocate Distribution and Target Libraries: DSNTIJAE

This job allocates three groups of data sets: the DB2 distribution library data sets, the DB2 target library data sets, and the SMP/E data sets. Depending on how your systems are set up, you might need to contact an MVS system programmer to help you manage some of the SMP/E data sets.

# Do not modify the data definition statements for SDSNCLST, SDSNLOAD, or  
# SDSNSAMP. The SDSNCLST, SDSNLOAD, and SDSNSAMP libraries must be  
# defined as partitioned data sets (PDS).

| For each group of data sets, DSNTIJAE **requires** a data set prefix, a volume name  
| on which to allocate the data sets, and a unit name (device type). These names are  
| called the *allocation job parameters*.

You need to change the allocation parameters according to the decisions you made regarding the LNKLST option, library definition, and SMP/E data set options. Table 41 on page 2-81 lists the allocation job parameters (prefix, volume, and unit name) for each of the three groups of data sets.



Table 41. SMP/E Allocation Job Parameters

| Data Sets                  | Parameter Type | Search Strings |
|----------------------------|----------------|----------------|
| DB2 Distribution Libraries | Prefix         | ?DLIBPRE?      |
|                            | Volume         | ?DLIBVOL?      |
|                            | Unit Name      | ?DLIBUNT?      |
| DB2 Target Libraries       | Prefix         | ?TARGPRE?      |
|                            | Volume         | ?TARGVOL?      |
|                            | Unit Name      | ?TARGUNT?      |
| SMP/E Control Data Sets    | Prefix         | ?SMPPRE?       |
|                            | TLIB Prefix    | ?TLIBPRE?      |
|                            | Volume         | ?SMPVOL?       |
|                            | TLIB Volume    | ?TLIBVOL?      |
|                            | Unit Name      | ?SMPUNIT?      |
|                            | Middle-level   | ?SMPMLQ?       |
|                            | Qualifier      |                |

If you are using JES3, you must split job DSNTIJAE into two jobs. A comment line in the code indicates where to split the job.

Examine the following items in the job you are using, and make any necessary modifications:

**Space Allocations:** The space allocated for the SMP/E history log data sets is rather large. The ddname for this data set is SMPLOG; its default name is IMSVS.HLDS. If you do not want to retain this log information, remove the data set allocations for ddnames SMPLOG and SMPLOGA from steps ALLOC and INITSMP of job DSNTIJAE. In step INITSMP, you will also need to specify *DA(NULLFILE)* in the DDDEF's for SMPLOG and SMPLOGA.

The ?TLIBVOL? parameter defines the location of the SMPTLIB data sets. The volume on which these data sets reside must have at least 35MB (1MB=1048576B) of free space. That is about 37 cylinders on a 3390 and 49 cylinders on a 3380.

The space allocations in DSNTIJAE for the CSI assume that you are using a 3380.

**SREL and DSSPACE:** The allocation jobs specify an SREL of P115 for the SMP/E data sets. Do **not** change this. They also specify DSSPACE to be (200,200,500). This is a minimum; change it only to increase it.

**SMP/E Zone Structure:** SMP/E zone structures are discussed in the *System Modification Program Extended (SMP/E) User's Guide*. You can choose to use a different zone structure from the one shown in DSNTIJAE.

---

## SMP/E Step 3: Run the RECEIVE Job: DSNTIJRC

Before you run the next three jobs, create backups of your DB2 for MVS/ESA Version 4 DB2 distribution and target libraries and your SMP/E data sets. You might need them if you have to fall back. If one of these three jobs fails, you probably need to delete and reallocate data sets or compress them before rerunning the job that failed. When rerunning one of these jobs, delete or comment out the parts that ran successfully, and rerun those that failed.

The SMP/E RECEIVE job, DSNTIJRC, loads the DB2 program modules, macros, and procedures into temporary data sets (SMPTLIBs). If this job fails or abends, correct the problem and rerun the job.

Examine the job before you run it. The ?SMPPRE? and ?SMPMLQ? parameters must have the same definition as in the allocation job. The SYSOUT class is defined as the same class as the job's MSGCLASS parameter.

At this point, you might wish to run an SMP/E APPLYCHECK to determine any service and any USERMODs that can be regressed by the following jobs.

If the IRLM (FMID HIR2101) on your system has had maintenance applied since it was shipped with DB2 or is a more current release than the one shipped with DB2, remove the HIR2101 step from job DSNTIJRC.

---

## SMP/E Step 4: Cleanup Job for Migration: DSNTIJUD (Optional)

We recommend that you avoid running this job by using new SMP/E zones for your migration. If this is not possible (you are installing DB2 for OS/390 Version 5 in the same SMP/E libraries in which you installed DB2 for MVS/ESA Version 4), you must run job DSNTIJUD.

Job DSNTIJUD is run for migration from Version 4 to ensure that delete processing is done properly before installing Version 5. It performs necessary SMP/E cleanup by deleting all Version 4 entries in the SMP/E target and distribution libraries. However, this job does not clean up the global zone. Issue the SMP/E REJECT command to remove entries from and clean up the global zone. DSNTIJUD should access the DSNTIJSM procedure shipped with Version 4.

You must use the IRLM (FMID HIR2101), which is shipped as part of DB2 Version 5, for both IMS and DB2. If you want to continue using the down-level IRLM for your IMS subsystem, you must have different IRLM levels in different SMP/E zones or SMP/E control data sets. Using the down-level IRLM increases your IRLM service activity and is not recommended.

Examine the job before you run it. The ?SMPPRE? and ?SMPMLQ? parameters must have the same definition as in the allocation job. The SYSOUT class is defined as the same class as the job's MSGCLASS parameter.

### Attention:

If DB2 shares the same CSI with any CICS or ISPF products, delete the following statements from this job before executing:

- DEL MOD(DFHEAI)
- DEL MOD(DFHEAIO)
- DEL MOD(ISPLINK)

DB2 may not share a CSI with CICS Version 4 because they both ship the DSNCRCT macro. Job DSNTIJUD should be run before the SMP/E APPLY (job DSNTIJAP). Running job DSNTIJUD is not necessary if you are installing DB2 for the first time. If you accidentally run it, it will have no adverse effect.

---

## SMP/E Step 5: Run the APPLY Job: DSNTIJAP

The SMP/E APPLY job, DSNTIJAP, copies and link-edits the DB2 program modules, macros, and procedures into the DB2 target libraries.

Examine the job before you run it. The ?SMPPRE? and ?SMPMLQ? parameters must have the same definition as in the allocation job. The SYSOUT class is defined as the same class as the job's MSGCLASS parameter.

The APPLY statement contains the CHECK parameter, which allows you to verify the APPLY without committing it. When you want to commit the APPLY, remove the CHECK parameter and rerun DSNTIJAP.

If you do not apply the FMIDs in a single APPLY statement as DSNTIJAP does, use the following order:

1. HIX5510, HIY5510, HIZ5510, and HDB5510 together
2. HIR2101.

Expect a return code of 4 from this job. Also expect link-edit error messages, which will cause link-edit return codes of 8. If you plan to use the DB2 Call Level Interface (CLI), see *Call Level Interface Guide and Reference* for information on the CLI FMID.

| Message  | Load Module | CSECTs          |
|----------|-------------|-----------------|
| IEW2454W | DSNCCOM0    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCCOM1    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCCOM2    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCEDF1    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCEDON    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCEXT1    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCEXT2    | DFHEI1, DFHEAI0 |
| IEW0461  | DSNCIAR     | DFHEI1, DFHEAI0 |
| IEW0461  | DSNTIAC     | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCMSG0    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCSM21    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCSM31    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCSTOP    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCSTRT    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNCUEXT    | DFHEI1, DFHEAI0 |
| IEW2454W | DSNHFT      | SQLCA1, SQLCA2  |
| IEW2470E | DSNCCOM0    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCCOM1    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCCOM2    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCEDF1    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCEDON    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCEXT1    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCEXT2    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCMSG0    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCSM21    | DFHEI1, DFHEAI0 |
| IEW2470E | DSNCSM31    | DFHEI1, DFHEAI0 |
| IEW2470E | DSNCSTOP    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCSTRT    | DFHEAI, DFHEAI0 |
| IEW2470E | DSNCUEXT    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCCOM0    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCCOM1    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCCOM2    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCEDF1    | DFHEAI, DFHEAI0 |

| Message  | Load Module | CSECTs          |
|----------|-------------|-----------------|
| GIM67301 | DSNCEDON    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCEXT1    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCEXT2    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNTIAC     | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCMSG0    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCSTM21   | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCSTM31   | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCSTOP    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCSTRT    | DFHEAI, DFHEAI0 |
| GIM67301 | DSNCUEXT    | DFHEAI, DFHEAI0 |

The errors for DFHEI1, DFHEAI, and DFHEAI0 are resolved later in the installation or migration process when you run job DSNTIJSU. The SQLCA1 and SQLCA2 references are resolved when DSNHFT is included in a FORTRAN application. If this job fails or abends, correct the problem and rerun the job.

If the IRLM (FMID HIR2101) on your system has had maintenance applied since it was shipped with DB2 or is a more current release than the one shipped with DB2, remove FMID HIR2101 from the APPLY step of job DSNTIJAP.

---

## SMP/E Step 6: Run the ACCEPT Job: DSNTIJAC

The SMP/E ACCEPT job, DSNTIJAC, copies the program modules, macros, and procedures into the DB2 distribution libraries. This allows you to apply corrective service later. If you do not want the DB2 components copied into the distribution libraries, do not run job DSNTIJAC.

Examine the job before you run it. The ?SMPPRE? and ?SMPMLQ? parameters must have the same definition as in the allocation job. The SYSOUT class is defined as the same class as the job's MSGCLASS parameter.

The ACCEPT statement contains the CHECK parameter, which allows you to verify the ACCEPT without committing it. When you want to commit the ACCEPT, remove the CHECK parameter and rerun DSNTIJAC5.

If this job fails or abends, correct the problem and rerun the job.

If the IRLM (FMID HIR2101) on your system has had maintenance applied since it was shipped with DB2 or is a more current release than the one shipped with DB2, remove FMID HIR2101 from the ACCEPT step of job DSNTIJAC.

---

## SMP/E Step 7: Unload the Jobs for the DB2I Panels

If you plan to use DB2I panels, you must choose between English panels and Kanji panels. To use Kanji DB2I panels, you need a terminal that can handle DBCS data, such as the IBM Personal System/55 workstation. You also need to make sure your ISPF environment is set up to display DBCS panels.

Use the packaging information in Table 42 on page 2-85 and the sample JCL in Figure 21 on page 2-85 as a guide. Make sure you specify the correct VOLSER, LABEL information, and data set names.

| <i>Table 42. Packaging information for the English and Kanji panels</i> |                                                                                                                                        |                     |
|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|---------------------|
|                                                                         | <b>English Panels</b>                                                                                                                  | <b>Kanji Panels</b> |
| <b>VOLSER</b>                                                           | DB5514                                                                                                                                 | DB5511              |
| <b>LABEL</b>                                                            | (2,SL)                                                                                                                                 | (2,SL)              |
| <b>Data set name</b>                                                    | IBM.JDB5514.F1                                                                                                                         | IBM.JDB5511.F1      |
| DSNTEJxx                                                                | DSNTNJxx specify a DSNTEJUD or a DSNTNJUD member to unload. No such member exists because no cleanup job is necessary for DB2I panels. |                     |

```

//* COMPID: DB2,5740XYR00
//* DOC: LOAD ENGLISH SMP INSTALLATION JCL FROM TAPE FOR DB2
//LOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//JCLTAPE DD DSN=IBM.JDB5514.F1,VOL=(PRIVATE,,SER=DB5514),
// UNIT=TAPE,LABEL=(2,SL),DISP=(OLD,PASS)
//*
//JCLDISK DD DSN=SYSADM.JCL.CNTL,VOL=SER=USER01,UNIT=SYSDA,
// DISP=OLD
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
 COPY I=JCLTAPE,0=JCLDISK
 SELECT MEMBER=(DSNTEJAE)
 SELECT MEMBER=(DSNTEJAC,DSNTEJAP,DSNTEJRC)
//*

```

*Figure 21. Sample JCL for Copying English Jobs to DASD*

---

## **SMP/E Step 8: Allocate Libraries for DB2I Panels**

To use English DB2I panels, run job DSNTEJAE. To use Kanji DB2I panels, run job DSNTNJAE. Use “SMP/E Step 2: Allocate Distribution and Target Libraries: DSNTIJAE” on page 2-80 as a guide to help you with this job.

---

## **SMP/E Step 9: Run the RECEIVE Job for the DB2I Panels**

To use English DB2I panels, run job DSNTEJRC. To use Kanji DB2I panels, run job DSNTNJRC. Use “SMP/E Step 3: Run the RECEIVE Job: DSNTIJRC” on page 2-81 as a guide to help you with this job.

---

## **SMP/E Step 10: Run the APPLY Job for the DB2I Panels**

To use English DB2I panels, run job DSNTEJAP. To use Kanji DB2I panels, run job DSNTNJAP. Use “SMP/E Step 5: Run the APPLY Job: DSNTIJAP” on page 2-83 as a guide to help you with this job.

---

## **SMP/E Step 11: Run the ACCEPT Job for the DB2I Panels**

To use English DB2I panels, run job DSNTJAC. To use Kanji DB2I panels, run job DSNTNJAC. Use “SMP/E Step 6: Run the ACCEPT Job: DSNTIJAC” on page 2-84 as a guide to help you with this job.

---

## **# SMP/E Step 12: Copy and Edit the SMP/E Jobs for DB2 REXX Language Support**

# You need to perform this step and the following three steps only if you are installing the DB2 REXX Language Support feature.

# Use the sample JCL shown in Figure 22 on page 2-87 to invoke the MVS utility IEBCOPY to copy the SMP/E jobs to DASD.

# After you have copied the SMP/E jobs to DASD, add a job statement to each job and customize the jobs to specify the unit names and volume serial numbers that your site uses.

# The SMP/E jobs move all files for DB2 REXX Language Support to the target and distribution libraries for the DB2 base product. Therefore, you do not need to set up target and distribution libraries for DB2 REXX Language Support.

---

## **# SMP/E Step 13: Run the RECEIVE Job for DB2 REXX Language Support**

# DSNTTJRC invokes SMP/E to receive the FMIDs for DB2 REXX Language Support into the SMP/E control data sets.

---

## **# SMP/E Step 14: Run the APPLY Job for DB2 REXX Language Support**

# DSNTTJAP invokes SMP/E to apply the FMIDs for DB2 REXX Language Support to the DB2 target libraries.

---

## **# SMP/E Step 15: Run the ACCEPT Job for DB2 REXX Language Support**

# DSNTTJAC invokes SMP/E to accept the FMIDs for DB2 REXX Language Support into the DB2 distribution libraries.

---

## **| SMP/E Step 16: Ensure Installation of Proper Maintenance**

| If you are migrating, your DB2 for MVS/ESA Version 4 subsystem must be at the proper maintenance level *before* migrating to DB2 for OS/390 Version 5. Refer to *DB2 Program Directory* for information on that proper maintenance level.

```

/* COMPID: DB2,5740XYR00
/* DOC: LOAD REXX SMP INSTALLATION JCL FROM TAPE FOR DB2
//LOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//JCLTAPE DD DSN=IBM.JDB551H.F1,VOL=(PRIVATE,,SER=DB551H),
// UNIT=TAPE,LABEL=(2,SL),DISP=(OLD,PASS)
/*
//JCLDISK DD DSN=SYSADM.JCL.CNTL,VOL=SER=USER01,UNIT=SYSDA,
// DISP=OLD
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
COPY I=JCLTAPE,O=JCLDISK
SELECT MEMBER=(DSNTTJAC,DSNTTJAP,DSNTTJRC)
/*
Figure 22. Sample JCL to Copy SMP/E jobs to DASD

```

---

## Finishing SMP/E Processing

Each of the display language control techniques described below is a way to set or change the current allocation of the ddname ISPLIB. If an ISPPALT allocation exists, then ISPF will use it instead of an ISPLIB allocation.

**Logon Procedures:** To switch languages, you need only to change the data set allocation currently in effect under the standard ISPF panel library ddname. A user's logon procedure can allocate ddname ISPLIB to select the current display language. Following is an example of a logon procedure:

```

/* THIS VERSION DISPLAYS ENGLISH PANELS */
//ISPLIB DD DSN=prefix.SDSNSPFP,DISP=SHR ENGLISH TASK
// DD DSN=prefix.SDSNPFPE,DISP=SHR ENGLISH DB2I

/* THIS VERSION DISPLAYS JAPANESE PANELS */
//ISPLIB DD DSN=prefix.SDSNSPFP,DISP=SHR ENGLISH TASK
// DD DSN=prefix.SDSNPFPK,DISP=SHR KANJI

```

**Language-switching CLISTs:** An ordinary CLIST can be used (outside of ISPF) to free and reallocate ISPLIB. Following is an example of a CLIST:

```

/* Execute this CLIST outside of ISPF */
PROC 0 LANGUAGE(E)
FREE DD(ISPLIB)
WRITE DO YOU WANT ENGLISH OR JAPANESE PANELS: Enter E or J.
READ &LANGUAGE;
IF &LANGUAGE = E +
 THEN ALLOC DD(ISPLIB) DS('DSN510.SDSNSPFP' 'DSN510.SDSNPFPE') +
 SHR /*ENGLISH*/
 ELSE ALLOC DD(ISPLIB) DS('DSN510.SDSNSPFP' 'DSN510.SDSNPFPK') +
 SHR /*JAPANESE*/
END

```

Some users allocate the ISPF panel library from their DEFAULT CLIST. Allocation of ddname ISPLIB controls the current language just as it does for the LOGON procedure.

**If you are falling back to DB2 for MVS/ESA Version 4**, change your logon procedures or CLISTs that use the Kanji feature to point to the Version 4 libraries.



---

## Chapter 2-5. Installing, Migrating, and Updating System Parameters

The values of parameters describe the operating characteristics of your DB2 system. You have to think of changing those values when **installing** DB2 or when **migrating** from DB2 for MVS/ESA Version 4 to DB2 for OS/390 Version 5. In between, you can consider **updating** them at any time to improve your operations.

When installing or migrating, you must run the installation CLIST to prepare jobs needed for later steps.

The installation CLIST displays a series of ISPF panels that prompt you to supply the parameter values or accept the defaults shown. The CLIST verifies that the values you enter are within the allowable ranges. The next section contains instructions for running the CLIST. The instructions identify the parameters and describe their purposes in the order in which they appear on the panels.

---

### Running the Installation CLIST

To use the ISPF panels, you must first make the DB2 ISPF library available to TSO and then invoke the installation CLIST in ISPF mode. You must be aware of the output that the panel session produces and take steps to save it for use later. Instructions for this procedure follow.

To use online help, you must:

- Set up online help according to the instructions in “Chapter 2-3. Setting Up DB2 Online Help” on page 2-51.
- Verify online help by entering BOOKMGR from the TSO command option.
- In your SYSPROC concatenation, make sure that the data set that contains the correct version of the DSNTINST CLIST is concatenated ahead of any other versions of DSNTINST.
- Invoke the DSNTINS0 CLIST.

If you do not want to use online help:

- Invoke the DSNTINST CLIST.

The installation CLIST allocates several data sets for input/output. From your TSO user ID, you should be able to allocate these data sets to the permanent or temporary unit names provided on installation panel DSNTIPA2. These devices may be defined by an esoteric device group. For more information on esoteric device groups, see *MVS/ESA MVS Configuration Program*.

### Making the DB2 ISPF Libraries Available to TSO

Concatenate the DB2 ISPF libraries to your normal allocations by issuing the following commands:

```

PROFILE WTP MSGID
ALLOCATE DDNAME(ISPMLIB) DSN('prefix.SDSNSPFM'
 'ISP.V3R5M0.ISPMENU' 'ISR.V3R5M0.ISRMENU') SHR REUSE
ALLOCATE DDNAME(ISPPLIB) DSN('prefix.SDSNSPFP'
 'ISP.V3R5M0.ISPPENU' 'ISR.V3R5M0.ISRPENU') SHR REUSE
ALLOCATE DDNAME(ISPSLIB) DSN('prefix.SDSNSPFS'
 'ISP.V3R5M0.ISPSLIB' 'ISR.V3R5M0.ISRSENU') SHR REUSE
ALLOCATE DDNAME(ISPTLIB) DSN('prefix.SDSNSPFT'
 'ISP.V3R5M0.ISPTLIB' 'ISR.V3R3M0.ISRTLIB') SHR REUSE

```

# The PROFILE command provides complete error messages. DB2 does not support  
# using LIBDEFs for the installation CLIST DSNTINS0 and online help.

The ALLOCATE command uses the default names of the libraries containing the ISPF panels. These ISPF library names might be different at your site. To concatenate or merge existing libraries with them, put the library names in the list of names in parentheses after DSN with the largest block size first. (If two or more libraries have the same block size, it does not matter which comes first.)

## Invoking the CLIST

1. Check your region size. Usually 2MB is enough.
2. Invoke ISPF.
3. Select option 6 on the main ISPF panel.
4. To use online help, enter:

```
EXEC 'prefix.SDSNCLST(DSNTINS0)'
```

Or, to receive messages tracing the CLIST progress,

```
EXEC 'prefix.SDSNCLST(DSNTINS0)' 'CONTROL(LIST)'
```

To NOT use online help, enter:

```
EXEC 'prefix.SDSNCLST(DSNTINST)'
```

Or, to receive messages tracing the CLIST progress,

```
EXEC 'prefix.SDSNCLST(DSNTINST)' 'CONTROL(LIST)'
```

## General Instructions

The CLIST reads a set of default values and displays them on the panels. The values can be either the original default values supplied by IBM or a set of values created by you in a previous CLIST run.

The installation CLIST saves the panel input into your DSNTIDxx output member just before the CLIST issues this message:

```
DSNT4781 BEGINNING EDITED DATA SET OUTPUT.
```

### Output from the Panel Session

As output, the panel session produces:

- A new data set member, if specified, that contains the parameter values resulting from the session. This member is stored in *prefix.SDSNSAMP*.
- A new data set, *prefix.NEW.SDSNSAMP*, that contains the edited JCL for later jobs and the values you enter on the panels

- A new data set, *prefix*.NEW.SDSNTEMP, containing tailored CLISTs for input to job DSNTIJVC, which is run during installation or migration.

You are asked to supply names, or accept defaults, for this output and supply the name for the load module for DB2 subsystem parameters (DSNZPxxx).

Figure 23 illustrates by examples how the CLIST functions.

Use job DSNTIJVC to combine the CLISTs into a common data set. If you are installing DB2, see “Making DB2 CLISTs Available to TSO and Batch Users: DSNTIJVC” on page 2-244 for more information. If you are migrating, see “Making DB2 CLISTs Available to TSO and Batch Users: DSNTIJVC” on page 2-276.

Some validity checking of the values you enter is done during the panel sessions. If you get an ISPF error message, press the HELP key for additional information.

*Figure 23. Examples of Input to and Output from the installation CLIST*

## Actions Allowed on Panels

All panel sequences begin with the Main Panel (DSNTIPA1), or with the Online Book Data Set Names panel (DSNTIPA0) if you are using online help.

**Preparation:** After the description of each parameter, record your choice for a value before you actually use the panels. If for some reason you exit the CLIST before you go through all the panels, your values are not saved.

Panels that have fields marked with asterisks show their values are primed on the basis of values from a previous panel. The following message is found on these panels:

```
DSNT444I SCROLLING BACKWARD MAY CHANGE FIELDS MARKED WITH ASTERISKS
```

If you scroll back to the panel which has the original value, the values on the succeeding panels are refreshed *only* if the original value is changed. If the values are changed, the following message is displayed:

```
DSNT443I VALUES MARKED WITH AN ASTERISK HAVE BEEN UPDATED
```

For example, panel DSNTIPH has fields marked with asterisks indicating values which are primed on the basis of the CATALOG ALIAS value (field 1) on installation panel DSNTIPA2.

**Help:** If you have Online Help installed, press the Help PF key (usually PF1) or enter HELP on the command line to get help for the choices you can make on each panel. Pressing the Help key takes you to a help menu panel. Type the number of the item for which you need help and press ENTER to link directly to detailed information on that subject in an online book. You can search or scroll through the book to find additional information on related topics as well. Press the EXIT PF key (usually PF3) to exit the book and return to the help menu panel. Press the END PF key (usually PF3) to return to the installation panels. You cannot make actual entries on help menu panels. Return to the installation panel to make the entry.

**Data Entry:** Enter your choice on a panel in the space marked by an arrow (====>). Begin your entry in the second position to the right of the arrow. (The first position is protected; you cannot write in it.)

**Panel IDs:** If you want the panel IDs to appear on each panel, enter the following command from any panel: PANELID ON.

## Reading the Panel Descriptions

**Scrolling Installation Panels:** The installation panels enable you to scroll back to previous panels to review or change values. The END key (usually PF3) will return you to the previous panel. Pressing ENTER continues to validate entries in the current panel and displays the next panel. If you want to exit completely from the installation process, use the RETURN key (usually PF4).

**Defaults:** The defaults shown in the text are the original defaults supplied by IBM. If you ran the CLIST before and saved the updated panel values in a DSNTIDxx data set, then the values you entered appear as defaults on the panels now. Panel values that are modified outside of the installation process are not saved in the output DSNTIDxx data set listed on panel DSNTIPA1 and are not on the panel.

**DSNHDECP Names:** These are the names of the parameters in the data-only load module DSNHDECP.

**Subsystem Parameter Names:** These are the names of the parameters in the data-only load module DSNZPxxx.

**Acceptable Values:** This part of the description gives you the range of allowable values or the list of allowable choices for an installation panel field—**not necessarily for the subsystem parameter value associated with that field.** If the maximum allowable value is over 1024, you can substitute that value for the equivalent K value. (The CLIST automatically multiplies the K value by 1024.) If the maximum allowable value is over 1 048 576, you can substitute that value for the equivalent M value. (The CLIST automatically multiplies the M value by 1 048 576.) The maximum acceptable values might be too large for smaller systems; therefore, make certain that the values you enter are valid for the size of your system.

**Update:** This information identifies a corresponding field on one of the update panels or refers to a page giving update instructions. When an option is specified, it refers to the option on the Update Selection Panel (DSNTIPB).

Your installation options are described in the sequence that DB2 presents the panels to you:

## Directory of Panels

Table 43 (Page 1 of 2). Panel Identifiers

| Panel ID | Panel Title                    | Page  |
|----------|--------------------------------|-------|
| DSNTIPA0 | Online Book Data Set Names     | 2-98  |
| DSNTIPA1 | Main Panel                     | 2-99  |
| DSNTIPA2 | Data Parameters                | 2-105 |
| DSNTIPK  | Define Group or Member         | 2-110 |
| DSNTIPH  | System Resource Data Set Names | 2-112 |
| DSNTIPT  | Data Set Names Panel 1         | 2-116 |
| DSNTIPU  | Data Set Names Panel 2         | 2-121 |
| DSNTIPQ  | Data Set Names Panel 3         | 2-127 |
| DSNTIPG  | Data Set Names Panel 4         | 2-130 |
| DSNTIPW  | Data Set Names Panel 5         | 2-133 |
| DSNTIPV  | CICS Release                   | 2-135 |
| DSNTIP3  | CICS Data Sets                 | 2-136 |
| DSNTIPD  | Sizes                          | 2-139 |

Table 43 (Page 2 of 2). Panel Identifiers

| Panel ID | Panel Title                                 | Page  |
|----------|---------------------------------------------|-------|
| DSNTIPE  | Thread Management                           | 2-145 |
| DSNTIP1  | Buffer Pool Sizes Panel 1                   | 2-150 |
| DSNTIP2  | Buffer Pool Sizes Panel 2                   | 2-152 |
| DSNTIPN  | Tracing                                     | 2-154 |
| DSNTIPO  | Operator Functions                          | 2-158 |
| DSNTIPF  | Application Programming Defaults Panel 1    | 2-164 |
| DSNTIP4  | Application Programming Defaults Panel 2    | 2-170 |
| DSNTIPI  | IRLM Panel 1                                | 2-174 |
| DSNTIPJ  | IRLM Panel 2                                | 2-177 |
| DSNTIPP  | Protection                                  | 2-181 |
| DSNTIPM  | MVS PARMLIB Updates                         | 2-186 |
| DSNTIPL  | Active Log Data Set Parameters              | 2-190 |
| DSNTIPA  | Archive Log Data Set Parameters             | 2-193 |
| DSNTIPS  | Databases and Spaces to Start Automatically | 2-199 |
| DSNTIPR  | Distributed Data Facility Panel 1           | 2-201 |
| DSNTIP5  | Distributed Data Facility Panel 2           | 2-205 |
| DSNTIPX  | Stored Procedures Parameters                | 2-207 |
| DSNTIPZ  | Data Definition Control Support             | 2-209 |
| DSNTIPY  | Job Editing                                 | 2-212 |
| DSNTIPC  | DB2 CLIST Calculations Panel 1              | 2-215 |
| DSNTIPC1 | DB2 CLIST Calculations Panel 2              | 2-219 |
| DSNTIPB  | Update Selection Menu                       | 2-228 |

## Directory of Panel Field Names

Table 44 (Page 1 of 5). Panel Fields

| Panel Field Name         | Panel    | Page  |
|--------------------------|----------|-------|
| ALLOCATION UNITS         | DSNTIPA  | 2-193 |
| APPL PROG AND SQL GUIDE  | DSNTIPA0 | 2-98  |
| APPL REGISTRATION TABLE  | DSNTIPZ  | 2-209 |
| APPLICATION DBRM         | DSNTIPT  | 2-116 |
| APPLICATION LOAD         | DSNTIPT  | 2-116 |
| ARCHIVE LOG FREQ         | DSNTIPL  | 2-190 |
| ARCHIVE LOG PW           | DSNTIPP  | 2-181 |
| ARCHIVE LOG RACF         | DSNTIPP  | 2-181 |
| ART/ORT ESCAPE CHARACTER | DSNTIPZ  | 2-209 |
| ASCII CODED CHAR SET     | DSNTIPF  | 2-164 |
| ASSEMBLER                | DSNTIPG  | 2-130 |
| ASSISTANT                | DSNTIPK  | 2-110 |
| AUDIT TRACE              | DSNTIPN  | 2-154 |
| AUTH MEMBER              | DSNTIPM  | 2-186 |
| AUTH SEQUENCE            | DSNTIPM  | 2-186 |
| AUTO BIND                | DSNTIPO  | 2-158 |
| AUTO START               | DSNTIPI  | 2-174 |
| BIND NEW PACKAGE         | DSNTIPP  | 2-181 |
| BLOCK SIZE               | DSNTIPA  | 2-193 |
| BP0 - BP29               | DSNTIP1  | 2-150 |
| BP30 - BP49              | DSNTIP2  | 2-152 |
| BP32K - BP32K9           | DSNTIP2  | 2-152 |
| BSDS PASSWORD            | DSNTIPP  | 2-181 |
| BUFFER POOL SIZE         | DSNTIPC  | 2-215 |
| C COMPILER               | DSNTIPU  | 2-121 |
| C COMPILER MESSAGES      | DSNTIPU  | 2-121 |
| C LIBRARY HEADERS        | DSNTIPU  | 2-121 |
| C LINK EDIT STUBS        | DSNTIPU  | 2-121 |
| C DYNAMIC RUNTIME        | DSNTIPU  | 2-121 |
| C PRE-LINK MESSAGES      | DSNTIPU  | 2-121 |
| C PROGRAM NAME           | DSNTIPU  | 2-121 |
| CACHE DYNAMIC SQL        | DSNTIP4  | 2-170 |
| CATALOG ALIAS            | DSNTIPA2 | 2-105 |

Table 44 (Page 2 of 5). Panel Fields

| Panel Field Name         | Panel    | Page  |
|--------------------------|----------|-------|
| CATALOG DATA             | DSNTIPA  | 2-193 |
| CHECKPOINT FREQ          | DSNTIPN  | 2-154 |
| CICS COBOL LIBRARY       | DSNTIP3  | 2-136 |
| CICS COBOL II LIBRARY    | DSNTIP3  | 2-136 |
| CICS LOAD LIBRARY        | DSNTIP3  | 2-136 |
| CICS MACRO LIBRARY       | DSNTIP3  | 2-136 |
| CICS PL/I LIBRARY        | DSNTIP3  | 2-136 |
| CICS RELEASE             | DSNTIPV  | 2-135 |
| CLIST LIBRARY            | DSNTIPT  | 2-116 |
| COBOL TYPE               | DSNTIPY  | 2-212 |
| CODE STORAGE SIZE        | DSNTIPC  | 2-215 |
| COLUMNS                  | DSNTIPD  | 2-139 |
| COMMAND PREFIX           | DSNTIPM  | 2-186 |
| COMMAND REFERENCE        | DSNTIPA0 | 2-98  |
| COMMAND SCOPE            | DSNTIPM  | 2-186 |
| COMPACT DATA             | DSNTIPA  | 2-193 |
| CONTROL ALL APPLICATIONS | DSNTIPZ  | 2-209 |
| COORDINATOR              | DSNTIPK  | 2-110 |
| COPY 1 NAME              | DSNTIPH  | 2-112 |
| COPY 2 NAME              | DSNTIPH  | 2-112 |
| COPY 1 PREFIX            | DSNTIPH  | 2-112 |
| COPY 2 PREFIX            | DSNTIPH  | 2-112 |
| CPP AUTO CALL LIBRARY    | DSNTIPU  | 2-121 |
| CPP CLASS LIBRARY        | DSNTIPU  | 2-121 |
| CPP CLASS LIB HEADERS    | DSNTIPU  | 2-121 |
| CPP COMPILER             | DSNTIPU  | 2-121 |
| CPP COMPILER MESSAGES    | DSNTIPU  | 2-121 |
| CPP DYNAMIC RUNTIME      | DSNTIPU  | 2-121 |
| CPP LINK EDIT STUBS      | DSNTIPU  | 2-121 |
| CPP PRE-LINK MESSAGES    | DSNTIPU  | 2-121 |
| CPP PROCEDURE LIBRARY    | DSNTIPG  | 2-130 |
| CPP PROGRAM NAME         | DSNTIPU  | 2-121 |
| CPP STANDARD HEADERS     | DSNTIPU  | 2-121 |
| CROSS MEMORY             | DSNTIPJ  | 2-177 |
| CSD NAME                 | DSNTIP3  | 2-136 |
| CURRENT DEGREE           | DSNTIP4  | 2-170 |
| DATABASES                | DSNTIPE  | 2-145 |
| DATABASES                | DSNTIPD  | 2-139 |
| DATA SET NAME(MEMBER)    | DSNTIPA1 | 2-99  |
| DATA SET STORAGE SIZE    | DSNTIPC  | 2-215 |
| DATA SHARING             | DSNTIPA1 | 2-99  |
| DATE FORMAT              | DSNTIP4  | 2-170 |
| DBRM LIBRARY             | DSNTIPT  | 2-116 |
| DB2 GENERIC LUNAME       | DSNTIPR  | 2-201 |
| DB2 GROUP                | DSNTIP3  | 2-136 |
| DB2 LOCATION NAME        | DSNTIPR  | 2-201 |
| DB2 NETWORK LUNAME       | DSNTIPR  | 2-201 |
| DB2 NETWORK PASSWORD     | DSNTIPR  | 2-201 |
| DB2 PROC NAME            | DSNTIPX  | 2-207 |
| DB2 SAMPLES GROUP        | DSNTIP3  | 2-136 |
| DDF STARTUP OPTION       | DSNTIPR  | 2-201 |
| DDF THREADS              | DSNTIPR  | 2-201 |
| DEADLOCK CYCLE           | DSNTIPJ  | 2-177 |
| DEADLOCK TIME            | DSNTIPJ  | 2-177 |
| DEALLOC PERIOD           | DSNTIPA  | 2-193 |
| DECIMAL ARITHMETIC       | DSNTIP4  | 2-170 |
| DECIMAL POINT IS         | DSNTIPF  | 2-164 |
| DECLARATION LIBRARY      | DSNTIPT  | 2-116 |
| DEFAULT INDEX TYPE       | DSNTIPE  | 2-145 |
| DEF ENCODING SCHEME      | DSNTIPF  | 2-164 |
| DEFINE CATALOG           | DSNTIPA2 | 2-105 |
| DEVICE TYPE 1            | DSNTIPA  | 2-193 |
| DEVICE TYPE 2            | DSNTIPA  | 2-193 |
| DIRECTORY/CATALOG        | DSNTIPP  | 2-181 |

Table 44 (Page 3 of 5). Panel Fields

| Panel Field Name          | Panel    | Page  |
|---------------------------|----------|-------|
| DIST SQL STR DELIMTR      | DSNTIPF  | 2-164 |
| DPROP SUPPORT             | DSNTIPO  | 2-158 |
| DRDA PORT                 | DSNTIP5  | 2-205 |
| DSMAX                     | DSNTIPC  | 2-215 |
| EBCDIC CODED CHAR SET     | DSNTIPF  | 2-164 |
| EDMPOOL STORAGE SIZE      | DSNTIPC  | 2-215 |
| EXECUTED STMTS            | DSNTIPD  | 2-139 |
| EXIT LIBRARY              | DSNTIPT  | 2-116 |
| EXPLAIN PROCESSING        | DSNTIPO  | 2-158 |
| EXTENDED SECURITY         | DSNTIPR  | 2-201 |
| FORTRAN COMPILER          | DSNTIPG  | 2-130 |
| FORTRAN LINK EDIT         | DSNTIPG  | 2-130 |
| GROUP ATTACH              | DSNTIPK  | 2-110 |
| GROUP NAME                | DSNTIPK  | 2-110 |
| IBM COBOL COMPILER        | DSNTIPQ  | 2-127 |
| IBM COBOL LINK EDIT       | DSNTIPQ  | 2-127 |
| IBM COBOL PRELINK MSGS    | DSNTIPQ  | 2-127 |
| IBM COBOL RUNTIME         | DSNTIPQ  | 2-127 |
| IBM COBOL SOM LIBRARY     | DSNTIPQ  | 2-127 |
| IDLE THREAD TIMEOUT       | DSNTIPR  | 2-201 |
| IMS RESLIB                | DSNTIPW  | 2-133 |
| INPUT BUFFER              | DSNTIPL  | 2-190 |
| INPUT MEMBER NAME         | DSNTIPA1 | 2-99  |
| INSTALL DD CONTROL SUPT.  | DSNTIPZ  | 2-135 |
| INSTALL IRLM              | DSNTIPI  | 2-174 |
| INSTALL TYPE              | DSNTIPA1 | 2-99  |
| INSTALLATION GUIDE        | DSNTIPA0 | 2-98  |
| IRLM LIBRARY              | DSNTIPT  | 2-116 |
| IRLM LOAD LIBRARY         | DSNTIPT  | 2-116 |
| IRLM LOCK MAXIMUM SPACE   | DSNTIPC  | 2-215 |
| IRLM XCF GROUP NAME       | DSNTIPJ  | 2-177 |
| IS DB2 GROUP NEW?         | DSNTIP3  | 2-136 |
| IS DB2 SAMPLES GROUP NEW? | DSNTIP3  | 2-136 |
| ISPF ISPLINK MODULE       | DSNTIPW  | 2-133 |
| LANGUAGE DEFAULT          | DSNTIPF  | 2-164 |
| LE/370 RUN TIME           | DSNTIPG  | 2-130 |
| LINK LIST ENTRY           | DSNTIPM  | 2-186 |
| LINK LIST LIBRARY         | DSNTIPT  | 2-116 |
| LINK LIST SEQUENCE        | DSNTIPM  | 2-186 |
| LIST NAME                 | DSNTIP3  | 2-136 |
| LOAD DISTRIBUTION         | DSNTIPT  | 2-116 |
| LOAD LIBRARY              | DSNTIPT  | 2-116 |
| LOCAL DATE LENGTH         | DSNTIP4  | 2-170 |
| LOCAL TIME LENGTH         | DSNTIP4  | 2-170 |
| LOCK ENTRY SIZE           | DSNTIPJ  | 2-177 |
| LOCKS PER TABLE(SPACE)    | DSNTIPJ  | 2-177 |
| LOCKS PER USER            | DSNTIPJ  | 2-177 |
| LOG PASSWORD              | DSNTIPP  | 2-181 |
| MACRO LIBRARY             | DSNTIPT  | 2-116 |
| MAX ABEND COUNT           | DSNTIPX  | 2-207 |
| MAX BATCH CONNECT         | DSNTIPE  | 2-145 |
| MAX KEPT DYN STMTS        | DSNTIPE  | 2-145 |
| MAX REMOTE ACTIVE         | DSNTIPE  | 2-145 |
| MAX REMOTE CONNECTED      | DSNTIPE  | 2-145 |
| MAX TSO CONNECT           | DSNTIPE  | 2-145 |
| MAX USERS                 | DSNTIPE  | 2-145 |
| MAXIMUM ECSA              | DSNTIPJ  | 2-177 |
| MEMBER IDENTIFIER         | DSNTIPJ  | 2-177 |
| MEMBER NAME               | DSNTIPK  | 2-110 |
| MINIMUM DIVIDE SCALE      | DSNTIPF  | 2-164 |
| MIXED DATA                | DSNTIPF  | 2-164 |
| MONITOR SIZE              | DSNTIPN  | 2-154 |
| MONITOR TRACE             | DSNTIPN  | 2-154 |
| MVS CATALOG               | DSNTIPP  | 2-181 |

Table 44 (Page 4 of 5). Panel Fields

| Panel Field Name         | Panel    | Page  |
|--------------------------|----------|-------|
| NUMBER OF COPIES         | DSNTIPH  | 2-112 |
| NUMBER OF LOGS           | DSNTIPL  | 2-190 |
| NUMBER TCBS              | DSNTIPX  | 2-207 |
| OBJT REGISTRATION TABLE  | DSNTIPZ  | 2-209 |
| OUTPUT BUFFER            | DSNTIPL  | 2-190 |
| OUTPUT MEMBER NAME       | DSNTIPA1 | 2-99  |
| PACKAGE AUTH CACHE       | DSNTIPP  | 2-181 |
| PACKAGE LISTS            | DSNTIPD  | 2-139 |
| PACKAGE STATEMENTS       | DSNTIPD  | 2-139 |
| PACKAGES                 | DSNTIPD  | 2-139 |
| PARAMETER MODULE         | DSNTIPO  | 2-158 |
| PERMANENT UNIT NAME      | DSNTIPA2 | 2-105 |
| PLAN AUTH CACHE          | DSNTIPP  | 2-181 |
| PLAN STATEMENTS          | DSNTIPD  | 2-139 |
| PLANS                    | DSNTIPD  | 2-139 |
| PL/I COMPILER            | DSNTIPG  | 2-130 |
| PL/I DYN RUNTIME BASE    | DSNTIPG  | 2-130 |
| PL/I DYN RUNTIME COMMON  | DSNTIPG  | 2-130 |
| PL/I LINK EDIT BASE      | DSNTIPG  | 2-130 |
| PL/I LINK EDIT COMMON    | DSNTIPG  | 2-130 |
| PREFIX                   | DSNTIPA1 | 2-99  |
| PRIMARY QUANTITY         | DSNTIPA  | 2-193 |
| PROC NAME                | DSNTIPI  | 2-174 |
| QUIESCE PERIOD           | DSNTIPA  | 2-193 |
| READ TAPE UNITS          | DSNTIPA  | 2-193 |
| RECALL DATABASE          | DSNTIPO  | 2-158 |
| RECALL DELAY             | DSNTIPO  | 2-158 |
| RECOMMENDED REAL STORAGE | DSNTIPC  | 2-215 |
| RECORDING MAX            | DSNTIPA  | 2-193 |
| REGISTRATION DATABASE    | DSNTIPZ  | 2-209 |
| REGISTRATION OWNER       | DSNTIPZ  | 2-209 |
| REMOTE LOCATION          | DSNTIPY  | 2-212 |
| REQUIRE FULL NAMES       | DSNTIPZ  | 2-209 |
| RESOURCE AUTHID          | DSNTIPP  | 2-181 |
| RESOURCE TIMEOUT         | DSNTIPI  | 2-174 |
| RESYNC INTERVAL          | DSNTIPR  | 2-201 |
| RESYNC PORT              | DSNTIP5  | 2-205 |
| RETENTION PERIOD         | DSNTIPA  | 2-193 |
| RID POOL SIZE            | DSNTIPC  | 2-215 |
| RLF AUTO START           | DSNTIPO  | 2-158 |
| RLST ACCESS ERROR        | DSNTIPO  | 2-158 |
| RLST ACCESS ERROR        | DSNTIPR  | 2-201 |
| RLST NAME SUFFIX         | DSNTIPO  | 2-158 |
| SAMPLE LIBRARY           | DSNTIPT  | 2-116 |
| SECONDARY QTY            | DSNTIPA  | 2-193 |
| SEQUENTIAL CACHE         | DSNTIPE  | 2-145 |
| SITE TYPE                | DSNTIPO  | 2-158 |
| SMF ACCOUNTING           | DSNTIPN  | 2-154 |
| SMF STATISTICS           | DSNTIPN  | 2-154 |
| SOM DLL IMPORT LIBRARY   | DSNTIPQ  | 2-127 |
| SORT LIBRARY             | DSNTIPW  | 2-133 |
| SORT POOL SIZE           | DSNTIPC  | 2-215 |
| SQL STRING DELIMITER     | DSNTIPF  | 2-164 |
| STATISTICS TIME          | DSNTIPN  | 2-154 |
| STD SQL LANGUAGE         | DSNTIP4  | 2-170 |
| STRING DELIMITER         | DSNTIPF  | 2-164 |
| SUBSYSTEM MEMBER         | DSNTIPM  | 2-186 |
| SUBSYSTEM NAME           | DSNTIPI  | 2-174 |
| SUBSYSTEM NAME           | DSNTIPM  | 2-186 |
| SUBSYSTEM SEQUENCE       | DSNTIPM  | 2-186 |
| SUFFIX                   | DSNTIPA1 | 2-99  |
| SYSTEM ADMIN 1           | DSNTIPP  | 2-181 |
| SYSTEM ADMIN 2           | DSNTIPP  | 2-181 |
| SYSTEM MACLIB            | DSNTIPW  | 2-133 |



Table 44 (Page 5 of 5). Panel Fields

| Panel Field Name              | Panel    | Page  |
|-------------------------------|----------|-------|
| SYSTEM OPERATOR 1             | DSNTIPP  | 2-181 |
| SYSTEM OPERATOR 2             | DSNTIPP  | 2-181 |
| SYSTEM PROCEDURES             | DSNTIPW  | 2-133 |
| TABLES                        | DSNTIPD  | 2-139 |
| TABLES IN STMT                | DSNTIPD  | 2-139 |
| TABLE SPACES                  | DSNTIPD  | 2-139 |
| TCP/IP ALREADY VERIFIED       | DSNTIP5  | 2-205 |
| TEMP CLIST LIBRARY            | DSNTIPT  | 2-116 |
| TEMP 4K DATA SETS             | DSNTIPD  | 2-139 |
| TEMP 4K SPACE                 | DSNTIPD  | 2-139 |
| TEMP 32K DATA SETS            | DSNTIPD  | 2-139 |
| TEMP 32K SPACE                | DSNTIPD  | 2-139 |
| TEMPORARY UNIT NAME           | DSNTIPA2 | 2-105 |
| TIME FORMAT                   | DSNTIP4  | 2-170 |
| TIME TO AUTOSTART             | DSNTIPI  | 2-174 |
| TIMEOUT VALUE                 | DSNTIPX  | 2-207 |
| TIMESTAMP ARCHIVES            | DSNTIPH  | 2-112 |
| TOTAL MAIN STORAGE            | DSNTIPC  | 2-215 |
| TOTAL STORAGE BELOW 16MB      | DSNTIPC  | 2-215 |
| TRACE AUTO START              | DSNTIPN  | 2-154 |
| TRACE SIZE                    | DSNTIPN  | 2-154 |
| U LOCK FOR RR/RS              | DSNTIPI  | 2-174 |
| UNKNOWN AUTHID                | DSNTIPP  | 2-181 |
| UNREGISTERED DDL DEFAULT      | DSNTIPZ  | 2-209 |
| UPDATE RATE                   | DSNTIPL  | 2-190 |
| UR CHECK FREQ                 | DSNTIPN  | 2-154 |
| USE PROTECTION                | DSNTIPP  | 2-181 |
| UTILITIES GUIDE AND REFERENCE | DSNTIPA0 | 2-98  |
| UTILITY CACHE OPTION          | DSNTIPE  | 2-145 |
| UTILITY TIMEOUT               | DSNTIPI  | 2-174 |
| VIEWS                         | DSNTIPD  | 2-139 |
| VOLUME SERIAL 1               | DSNTIPA2 | 2-105 |
| VOLUME SERIAL 2               | DSNTIPA2 | 2-105 |
| VOLUME SERIAL 3               | DSNTIPA2 | 2-105 |
| VOLUME SERIAL 4               | DSNTIPA2 | 2-105 |
| VOLUME SERIAL 5               | DSNTIPA2 | 2-105 |
| VOLUME SERIAL 6               | DSNTIPA2 | 2-105 |
| VOLUME SERIAL 7               | DSNTIPA2 | 2-105 |
| VS COBOL COMPILER             | DSNTIPQ  | 2-127 |
| VS COBOL LINK EDIT            | DSNTIPQ  | 2-127 |
| VS COBOL II COMPILER          | DSNTIPQ  | 2-127 |
| VS COBOL II LINK EDIT         | DSNTIPQ  | 2-127 |
| WLM PROC NAME                 | DSNTIPX  | 2-207 |
| WORK FILE DB                  | DSNTIPK  | 2-110 |
| WORKING STORAGE SIZE          | DSNTIPC  | 2-215 |
| WRITE THRESHOLD               | DSNTIPL  | 2-190 |
| WRITE TO OPER                 | DSNTIPA  | 2-193 |
| WTO ROUTE CODES               | DSNTIPO  | 2-158 |
| WTOR ROUTE CODE               | DSNTIPA  | 2-193 |

## Online Book Data Set Names Panel: DSNTIPA0

```
DSNTIPA0 ONLINE BOOK DATA SET NAMES
====> _

 Welcome to DB2 Version 5

You have invoked the DSNTINS0 CLIST, which enables the online
help. To use online help, you must set it up according to the
instructions in the IBM DATABASE 2 Program Directory. If you
have not set up online help, do so before continuing to run this
CLIST. If you do not want to use online help, exit now and invoke
the DSNTINST CLIST.

If you changed the data set names when you unloaded the DB2 books,
enter those names below. All book data set names must end with .BOOK.

 1 APPL PROG AND SQL GUIDE ===> DSNHELP.DSNAP0F1.BOOK
 2 COMMAND REFERENCE ===> DSNHELP.DSNCR0F1.BOOK
 3 INSTALLATION GUIDE ===> DSNHELP.DSNIG0F1.BOOK
 4 UTILITIES GUIDE AND REF ===> DSNHELP.DSNUG0F1.BOOK

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 24. Online Book Data Set Names Panel: DSNTIPA0

### 1. APPL PROG AND SQL GUIDE

Acceptable values: Valid data set name ending in .BOOK. See page 2-101.  
Default: DSNHELP.DSNAP0F1.BOOK  
DSNZPARAM: none

### 2. COMMAND REFERENCE

Acceptable values: Valid data set name ending in .BOOK. See page 2-101.  
Default: DSNHELP.DSNCR0F1.BOOK  
DSNZPARAM: none

### 3. INSTALLATION GUIDE

Acceptable values: Valid data set name ending in .BOOK. See page 2-101.  
Default: DSNHELP.DSNIG0F1.BOOK  
DSNZPARAM: none

### 4. UTILITIES GUIDE AND REF

Acceptable values: Valid data set name ending in .BOOK. See page 2-101.  
Default: DSNHELP.DSNUG0F1.BOOK  
DSNZPARAM: none

## Main Panel: DSNTIPA1

The entries on the Main Panel control input to and output from the installation CLIST. When processing is complete, this panel is displayed again. The values you enter are saved in the ISPF profile for your authorization ID and are displayed each time you run the CLIST.

You must specify an output member name in field 7 to save your panel input.

The DSNTINST CLIST saves the panel input into your DSNTIDxx output member just before the CLIST issues this message:

```
DSNT4781 BEGINNING EDITED DATA SET OUTPUT.
```

```
DSNTIPA1 INSTALL, UPDATE, MIGRATE DB2 - MAIN PANEL
====> _

Check parameters and reenter to change:

 1 INSTALL TYPE ====> INSTALL Install, Update, or Migrate
 2 DATA SHARING ====> NO Yes, No, or blank for Update

Enter the following value for migration only:
 3 DATA SET NAME(MEMBER)====>

Enter name of your input data sets (SDSNLOAD, SDSNMACS, SDSNSAMP, SDSNCLST):
 4 PREFIX ====> DSN510
 5 SUFFIX ====>

Enter to set or save panel values (by reading or writing the named members):

 6 INPUT MEMBER NAME ====> DSNTIDXA Enter to read old panel values
 7 OUTPUT MEMBER NAME ====> Enter to write new panel values

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 25. Main Panel: DSNTIPA1

### 1. INSTALL TYPE

Acceptable values: Install, Update, Migrate  
Default: INSTALL  
DSNZPxxx: none

Choose whether you are installing, updating, or migrating.

- Use **Install** to install DB2 for the first time. This is the default for the first run of the CLIST.
- Use **Update** to update parameters for an existing DB2 subsystem.
- Use **Migrate** to migrate from DB2 for MVS/ESA Version 4 to DB2 for OS/390 Version 5. You are required to fill in field 3 (DATA SET NAME(MEMBER)) when you are migrating.

If you are updating or migrating, you use the same set of panels you use for installation. Each panel displays all fields; however, the fields that cannot be

changed in update or migrate mode are protected. This way, you can see the values related to ones that you wish to change.

You can also choose either INSTALL or UPDATE to recheck values you chose before.

Certain fields cannot be changed during a migration. See page Figure 60 on page 2-229 for that information. Be sure those fields are correct in the data set member you provide.

## 2. DATA SHARING

Acceptable values: Yes, No, or blank for Update  
Default: No  
DSNZPxxx: DSN6GRP DSHARE

Specify whether you want to use the data sharing function. Choose **NO** if you are not using data sharing. If you choose YES, you will continue to panel DSNTIPK after completing panel DSNTIPA2.

**If you specify Yes during installation, this window appears:**

```
DSNTIPP1

DATA SHARING FUNCTION:

Select one.
 1. Group
 2. Member
 3. Enable

PRESS: ENTER to continue
 RETURN to exit
```

Figure 26. DSNTIPP1

### DATA SHARING FUNCTION:

Acceptable values: Group, Member, Enable  
Default: none  
DSNZPxxx: none

Specify a data sharing function. A value is required. After entering a value, you proceed to panels DSNTIPA2 and DSNTIPK. See *Data Sharing: Planning and Administration* for more information about the Group, Member, and Enable functions.

If you specify Yes during migration, a window displays asking if the current member is the first to migrate.

```

DSNTIPP2

FIRST MEMBER OF GROUP TO MIGRATE?

Select one.
 1. Yes
 2. No

PRESS: ENTER to continue
 RETURN to exit

```

Figure 27. DSNTIPP2

### FIRST MEMBER OF GROUP TO MIGRATE?

|                    |         |
|--------------------|---------|
| Acceptable values: | Yes, No |
| Default:           | none    |
| DSNZPxxx:          | none    |

Specify Yes if this is the first member of a data sharing group to migrate. A value is required.

After entering a value, you proceed to panels DSNTIPA2 and DSNTIPK.

### 3. DATA SET NAME(MEMBER)

|                    |                              |
|--------------------|------------------------------|
| Acceptable values: | 1-44 alphanumeric characters |
| Default:           | NULL                         |
| DSNZPxxx:          | none                         |

Specify the name of the input data set to use for migrating from DB2 for MVS/ESA Version 4. The member named contains the output parameters produced when you last installed, updated, or migrated DB2. Give the fully qualified data set name in the form:

any.data.set.name(member)

This is an example of an actual data set name:

DSN510.SDSNSAMP(DSNTIDXA)

If you no longer have this data set member, or the one you have is incorrect, use the installation or update process from your previous release to re-create or correct the member. Enter the correct values on the panels, and save them under a new output member name. Discard the JCL created by this process; use the newly created member for migration.

If you are installing or updating, the field must remain blank.

**Valid Data Set Name:** There are two forms of a valid data set name:

**Unqualified Name:** 1 to 8 alphanumeric or national characters, a hyphen, or the character X'C0'. The first character must be alphabetic or national. Do not

use hyphens in data set names for RACF-protected data sets. For example, ALPHA is an unqualified data set name.

**Qualified Name:** Multiple names joined by periods. Each name is coded like an unqualified name. Therefore, the name must contain a period every eight characters or less. For example, ALPHA.PGM is a qualified data set name. The maximum length of a qualified data set name is:

- 44 characters if you use the TSO PROFILE setting NOPREFIX.
- 42 characters if you use the TSO PROFILE setting PREFIX.
- For an output tape data set, 17 characters, including periods. If the name is longer than 17 characters, only the rightmost 17 characters are written to the tape header label.

#### 4. PREFIX

|                    |                 |
|--------------------|-----------------|
| Acceptable values: | 1-18 characters |
| Default:           | DSN510          |
| DSNZPxxx:          | none            |

Specify the input prefix for the SDSNLOAD, SDSNMACS, SDSNSAMP, SDSNDBRM and SDSNCLST libraries. The prefix must be the same as the name you specified for the symbolic parameter TARGPRE in SMP/E job DSNTIJAE.

This is also the prefix for several partitioned data sets which are deleted if they exist, and are created during the tailoring session. If you use the default DSNTIDXA in field 6 (INPUT MEMBER NAME), the prefix for fields 1, 2, 3, and the prefix and suffix for fields 4 through 11 on installation panel DSNTIPT on page 2-116 are set. If all these data sets do not have the same prefix and suffix, you can change them on installation panel DSNTIPT.

#### 5. SUFFIX

|                    |                 |
|--------------------|-----------------|
| Acceptable values: | 0-17 characters |
| Default:           | NULL            |
| DSNZPxxx:          | none            |

Specify a suffix to the names listed below. The fully qualified data set name cannot exceed 44 characters. Names exceeding eight characters must be in groups of no more than eight characters, separated by periods.

Use a suffix only if you have added a common suffix to the following libraries when you created them in job DSNTIJAE:

|                         |                         |                         |
|-------------------------|-------------------------|-------------------------|
| <i>prefix</i> .ADSNLOAD | <i>prefix</i> .SDSNLINK | <i>prefix</i> .SDSNMACS |
| <i>prefix</i> .SDSNLOAD | <i>prefix</i> .SDSNEXIT | <i>prefix</i> .ADSNMACS |
| <i>prefix</i> .SDSNCLST | <i>prefix</i> .SDSNSAMP | <i>prefix</i> .SDSNDBRM |
| <i>prefix</i> .SDXRRESL |                         |                         |

If you did not add a common suffix to these libraries, be sure to enter their correct data set names on panel DSNTIPT.

To use the default DB2 data set names, use *DSN510* in field 4, and leave field 5 blank.

## 6. INPUT MEMBER NAME

Acceptable values: 1-8 characters  
Default: DSNTIDXA  
DSNZPxxx: none

Specify the input member name of the data set that contains the default parameter values for installing and migrating, as in *prefix.SDSNSAMP.suffix*. To install DB2 for the first time, use the IBM-supplied defaults in member DSNTIDXA. If you process the panels several times within a single run of the CLIST, all the previous values entered, except edited output data sets, will remain the same.

For migration, give two member names for input values: one in the INPUT MEMBER NAME field, and one in the DATA SET NAME(MEMBER) (field 3). The INPUT MEMBER NAME must specify a member that contains the default parameter values for the new release (usually DSNTIDXA) and is applied first to establish the CLIST parameters. However, if you are migrating a second or subsequent data sharing member to Version 5 then the INPUT MEMBER NAME is the OUTPUT MEMBER NAME used when migrating the first member of the data sharing group to Version 5. The DATA SET NAME(MEMBER) field must specify a member containing the DB2 for MVS/ESA Version 4 values at your site. This member is applied last and overrides the CLIST values established by the member specified in field 6.

To install DB2 using parameters from a previous run as defaults, you must supply the member that contains the output from the previous run. It was the OUTPUT MEMBER NAME (field 7) last time.

The following data set names are generated with the *prefix* and *suffix* values from fields 4 and 5 only when the input member name for field 6 is DSNTIDXA.

Table 45. Resulting Data Set Names When Using Prefix and Suffix Parameters

| Default Library Name        | CLIST Edited Library Name                  |
|-----------------------------|--------------------------------------------|
| <i>prefix</i> .DBRMLIB.DATA | <i>prefix</i> .DBRMLIB.DATA. <i>suffix</i> |
| <i>prefix</i> .RUNLIB.LOAD  | <i>prefix</i> .RUNLIB.LOAD. <i>suffix</i>  |
| <i>prefix</i> .SRCLIB.DATA  | <i>prefix</i> .SRCLIB.DATA. <i>suffix</i>  |
| <i>prefix</i> .SDSNDBRM     | <i>prefix</i> .SDSNDBRM. <i>suffix</i>     |
| <i>prefix</i> .SDSNLINK     | <i>prefix</i> .SDSNLINK. <i>suffix</i>     |
| <i>prefix</i> .SDSNLOAD     | <i>prefix</i> .SDSNLOAD. <i>suffix</i>     |
| <i>prefix</i> .SDSNMACS     | <i>prefix</i> .SDSNMACS. <i>suffix</i>     |
| <i>prefix</i> .ADSNLOAD     | <i>prefix</i> .ADSNLOAD. <i>suffix</i>     |
| <i>prefix</i> .ADSNMACS     | <i>prefix</i> .ADSNMACS. <i>suffix</i>     |
| <i>prefix</i> .SDSNSAMP     | <i>prefix</i> .SDSNSAMP. <i>suffix</i>     |
| <i>prefix</i> .SDSNCLST     | <i>prefix</i> .SDSNCLST. <i>suffix</i>     |
| <i>prefix</i> .SDXRRESL     | <i>prefix</i> .SDXRRESL. <i>suffix</i>     |

## 7. OUTPUT MEMBER NAME

Acceptable values: 1-8 characters  
Default: NULL  
DSNZPxxx: none

Specify the member name of the output data set in which to save the values you enter on the panels. If you give no name, the values are lost when you leave the

installation CLIST. This member is stored in *prefix*.SDSNSAMP (not the one created by the DSNTINST CLIST). To avoid replacing any members of *prefix*.SDSNSAMP that were shipped with the product, specify DSNTIDxx as the value of OUTPUT MEMBER NAME, where xx is any alphanumeric value except XA or VB.

Always give a new value in the OUTPUT MEMBER NAME field for a new panel session. You supply the name from your current session in the INPUT MEMBER NAME field for your next session. You should not use the same member name for output as for input. You could find it convenient to write down the output member name entered here for reference during future sessions.

## Recommended Approach for a New Installer

If you are a first-time installer, try the following suggestions:

- For field 1, INSTALL TYPE, enter Install.
- Set fields 4 and 5, PREFIX and SUFFIX, to the values you used when you allocated the DB2 libraries using job DSNTIJAE.
- In the INPUT MEMBER NAME field, use DSNTIDXA (the default) for the first run. For any later runs, the CLIST sets the default input name to the prior output name.
- Specify a value in OUTPUT MEMBER NAME (field 7) only when you want your options saved. Specify values in TEMP CLIST LIBRARY (field 1), CLIST LIBRARY (field 3), and SAMPLE LIBRARY (field 2) on installation panel DSNTIPT on 2-116 only when you want output data sets tailored.

Do not run the installation jobs before tailoring them with the CLIST. If you later want to update the parameters, the CLIST sets the default input name to the prior output name.



## Data Parameters Panel: DSNTIPA2

The entries on this panel define the DASD volumes for the subsystem databases and data sets. The values you enter on this and each of the following panels are saved in the data set member you named in the OUTPUT MEMBER NAME field on the Main Panel.

For information on updating parameters with changes that cannot be made using the panels, see "The Update Process" on page 2-228.

```
DSNTIPA2 INSTALL DB2 - DATA PARAMETERS
===> -

Check parameters and reenter to change:

 1 CATALOG ALIAS ===> DSNCAT Alias of VSAM catalog for
 DB2 subsystem data sets
 2 DEFINE CATALOG ===> YES YES or NO
 3 VOLUME SERIAL 1 ===> DSNV01 CLIST allocation
 4 VOLUME SERIAL 2 ===> DSNV01 Non-VSAM data
 5 VOLUME SERIAL 3 ===> DSNV02 VSAM catalog, default, and
 work file database
 6 VOLUME SERIAL 4 ===> Directory, catalog data
 7 VOLUME SERIAL 5 ===> Directory, catalog indexes
 8 VOLUME SERIAL 6 ===> Log copy 1, BSDS 2
 9 VOLUME SERIAL 7 ===> Log copy 2, BSDS 1
10 PERMANENT UNIT NAME ===> 3390 Device type for MVS catalog
 and partitioned data sets
11 TEMPORARY UNIT NAME ===> SYSDA Device type for
 temporary data sets

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 28. Data Parameters Panel: DSNTIPA2

### 1. CATALOG ALIAS

Acceptable values: 1-8 characters  
Default: DSNCAT  
Update: see Section 2 (Volume 1) of *Administration Guide*  
DSNZPxxx: DSN6SPRM CATALOG

Specify the alias of the VSAM integrated catalog facility user catalog or the name of the VSAM integrated catalog facility master catalog in which to catalog the DB2 VSAM data sets created during installation.

**VSAM Data Set Cataloging Options:** The installation jobs catalog the DB2 VSAM data sets. This includes recovery log, subsystem, and user data sets. You must create the catalog that defines these data sets through the VSAM integrated catalog facility. We recommend that you use an integrated catalog facility user catalog to catalog all DB2 objects because you can use aliases for user catalogs. When you use the CREATE STOGROUP statement, you might need to use an alias for the VCAT option, which must be a single-level 1 to 8 character name. You can use a master catalog, but only if the name of the master catalog is a single-level name of 1 to 8 characters.

Be sure that your alias conforms to your local naming conventions. To change this parameter for a previously installed DB2 subsystem, see Section 2 (Volume 1) of *Administration Guide*. It is best to use the same integrated catalog facility catalog alias for DB2 for OS/390 Version 5 that you used for DB2 for MVS/ESA Version 4 because Version 5 uses many of your Version 4 data sets that are already cataloged.

Whether you are installing or migrating, DB2 does not require you to catalog all DB2 VSAM data sets in the same integrated catalog facility catalog. We call the catalog that you create when installing the *primary* integrated catalog facility catalog. You must catalog some data sets in the primary catalog. You can catalog other data sets elsewhere, and some data sets need not be cataloged at all. See Table 46 for a list of available options.

Table 46. DB2 Data Sets Integrated Catalog Facility Catalog Options

| DB2 Data Sets                                                                                                                    | Options                                                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DB2 directory (DSNDB01) (VSAM)<br>DB2 catalog (DSNDB06) (VSAM)<br>Default database (DSNDB04) (VSAM)<br>Work file database (VSAM) | You must catalog these data sets in the primary integrated catalog facility catalog.                                                                                                           |
| Active logs (VSAM)<br>Bootstrap data set (VSAM)                                                                                  | You can catalog these in a different integrated catalog facility catalog and give them a prefix different from those in the primary catalog.                                                   |
| Archive logs (QSAM)                                                                                                              | If the archive log data set is allocated on DASD, then the data set must be cataloged. If the archive log data set is allocated on a tape device, you have the option to catalog the data set. |
| User table spaces<br>User index spaces                                                                                           | You need not put these in the primary catalog. You can put different user spaces in different integrated catalog facility catalogs.                                                            |

You must provide any catalog connections for log and bootstrap data sets that you do not catalog on the primary DB2 integrated catalog facility catalog. We suggest that you add an alias for the proper catalog.

Although you can catalog the two DB2 subsystems on the same integrated catalog facility catalog, they must not share the same integrated catalog facility catalog alias, because this is the only parameter that makes the data set names unique.

**Data Set Naming Conventions:** The value you specify as the MVS catalog alias is also used as the high-level qualifier for DB2 VSAM data sets. The data sets for the DB2 directory and catalog databases, the default database, and the temporary database are all VSAM linear data sets (LDSs). Their data set names have the following format:

```
ddddddd.DSNDBn.bbbbbbb.xxxxxxx.I0001.Accc
```

where:

ddddddd Is the high-level qualifier, the value you supply for this field.

**DSNDBn** Is a constant identifying this as a DB2 data set; *n* is C for a cluster name or D for a data component name.

*bbbbbbbb* Is the database name. The system database names are:  
**DSNDB01** The DB2 directory database  
**DSNDB04** The default database  
**DSNDB06** The DB2 catalog database  
**DSNDB07** The work file database

*xxxxxxx* Is the name of the individual table space or index space.

**I0001.Accc** Identifies the data set. *ccc* is the partition number of a partitioned table space or index space, or the relative data set number of a simple or segmented table space or index space.

For example, if the catalog alias is DSNCAT, one of the DB2 directory data sets is named:

DSNCAT.DSNDBD.DSNDB01.DBD01.I0001.A001

Similarly, one of the DB2 catalog data sets is named:

DSNCAT.DSNDBD.DSNDB06.SYSDBASE.I0001.A001

## 2. DEFINE CATALOG

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| Acceptable values: | YES, NO                                                 |
| Default:           | YES                                                     |
| Update:            | see Section 2 (Volume 1) of <i>Administration Guide</i> |
| DSNZPxxx:          | none                                                    |

Specify whether you want to create a new integrated catalog facility catalog. YES builds a new integrated catalog facility catalog using the alias you gave in field 1. NO signals that the catalog named by field 1 already exists; the CLIST does not create a new one.

If you specify YES, DB2 creates a user catalog and an alias for that catalog. DB2 creates the high-level qualifier of the catalog name by adding a number to the end of the alias you defined in field 1. If the alias has less than eight characters, DB2 appends a 1 to the end. For example, if you accepted the default of DSNCAT for field 1, the catalog created is named DSNCAT1.USER.CATALOG. If the alias has eight characters, DB2 changes the last character into a 1. If the last character is already a 1, DB2 changes the 1 into a 2.

### 3-9. VOLUME SERIAL 1– VOLUME SERIAL 7

|                    |                                  |
|--------------------|----------------------------------|
| Acceptable values: | any valid MVS volume serial name |
| Default:           | DSNV01–DSNV02                    |
| Update:            | see Update on page 2-228         |
| DSNZPxxx:          | none                             |

#  
#  
#  
#  
#  
#

Specify the volume serial numbers for the data sets defined by installation or migration. Each volume serial number must identify a volume that is accessible to MVS for dynamic allocation of the data set, and all identified volumes must be of the same device type and have adequate free space. See Section 2 (Volume 1) of *Administration Guide* for more information on managing data sets. Refer to “Chapter 2-2. Estimating DB2 Storage Needs” on page 2-15 for information on

# estimating space. You cannot use an asterisk (\*) as a pattern-matching character in this field—you must use a volume serial number.

Specifying more than one volume for VOLUME SERIAL 1 through VOLUME SERIAL 7 helps recovery and performance. A series of messages are produced that estimate space distribution on the volumes specified. See “Install DB2—CLIST Calculations Panel 2: DSNTIPC1” on page 2-219 for examples. If you specify fewer than six volumes, data is combined on them. If you specify six volumes, data is distributed as follows:

| <b>Field</b>                                     | <b>Is used for ...</b>                                                                                                                                                                                                                                                                   |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (VOLUME SERIAL 1)<br>(Field 3 on Panel DSNTIPA2) | CLIST allocation. Only this volume is required for the tailoring session. VOLUME SERIAL 1 through VOLUME SERIAL 7 are not required until you begin running the tailored jobs. The default is DSNV01. This volume is used for <i>prefix.NEW.SDSNTEMP</i> and <i>prefix.NEW.SDSNSAMP</i> . |
| (VOLUME SERIAL 2)<br>(Field 4 on Panel DSNTIPA2) | Non-VSAM data. It is used for <i>prefix.DBRMLIB.DATA</i> , <i>prefix.RUNLIB.LOAD</i> , and <i>prefix.SRCLIB.DATA</i> . The default is DSNV01.                                                                                                                                            |
| (VOLUME SERIAL 3)<br>(Field 5 on Panel DSNTIPA2) | The temporary data sets, the default and sample storage group, and the VSAM catalog (if a new one is created). The default is DSNV02.                                                                                                                                                    |
| (VOLUME SERIAL 4)<br>(Field 6 on Panel DSNTIPA2) | DB2 catalog and directory. If you leave this field blank, it is set to the value you gave for field 4.                                                                                                                                                                                   |
| (VOLUME SERIAL 5)<br>(Field 7 on Panel DSNTIPA2) | DB2 catalog indexes and directory indexes. If you leave this field blank, it is set to the value you gave for field 5.                                                                                                                                                                   |
| (VOLUME SERIAL 6)<br>(Field 8 on Panel DSNTIPA2) | The first copy of the active log and the second copy of the bootstrap data set (BSDS). If you leave this field blank, it is set to the value you gave for field 6.                                                                                                                       |
| (VOLUME SERIAL 7)<br>(Field 9 on Panel DSNTIPA2) | The second copy of the active log and the first copy of the BSDS. If you leave this field blank, it is set to the value you gave for field 7 on this panel.                                                                                                                              |

If you accept the default for dual active logging, give different volume serials for field 8 and field 9. That places the active logs on different DASD devices.

During migration you can change volume serial 1, but you cannot change volume serials 2-7.

## 10. PERMANENT UNIT NAME

|                    |                                |
|--------------------|--------------------------------|
| Acceptable values: | valid device type or unit name |
| Default:           | 3390                           |
| Update:            | see Update on page 2-228       |
| DSNZPxxx:          | none                           |

Specify the device type or unit name that will be used to allocate the following data sets:

Integrated catalog facility catalog

*prefix.DBRMLIB.DATA.suffix*

*prefix.RUNLIB.LOAD.suffix*

*prefix.SRCLIB.DATA.suffix*

The two data sets generated by the DSNTINST CLIST

- *prefix.NEW.SDSNTEMP*
- *prefix.NEW.SDSNSAMP*

The value identifies a direct access unit name for partitioned data sets and the integrated catalog facility catalog. If you want to use different device types for different data sets, edit the installation or migration jobs after you complete the tailoring session. Some common device types are 3380, 3390, and 9340.

The value is sometimes used during IVP processing to place output (from COPY TABLESPACE, for example) on the device type specified here.

A change to this parameter during migration does not affect the integrated catalog facility catalog, DB2 catalog, directory, or logs. The new value is used for data sets created during migration.

## 11. TEMPORARY UNIT NAME

|                    |                                |
|--------------------|--------------------------------|
| Acceptable values: | valid device type or unit name |
| Default:           | SYSDA                          |
| Update:            | see Update on page 2-228       |
| DSNZPxxx:          | none                           |

Specify the device type or unit name for allocating temporary data sets. It is the direct access or disk unit name used for the precompiler, compiler, assembler, sort, linkage editor, and utility work files in the tailored jobs and CLISTs.

## Define Group or Member Panel: DSNTIPK

This panel follows panel DSNTIPA2 when you select a data sharing function (GROUP, MEMBER, or ENABLE). You must start DB2 and IRLM group names with a letter. You should carefully consider the naming convention for a data sharing system. See *Data Sharing: Planning and Administration* for guidance on planning a naming convention before you choose names for the fields on panel DSNTIPK.

```
DSNTIPK INSTALL DB2 - DEFINE GROUP OR MEMBER
====> _

Check parameters and reenter to change:

 1 GROUP NAME ====> DSNCAT Name of the DB2 group
 2 MEMBER NAME ====> DSN1 Name of DB2 member in group
 3 WORK FILE DB ====> DSN1 Work file database name for this member
 4 GROUP ATTACH ====> Group attach name for TSO, batch, utilities
 5 COORDINATOR ====> NO NO or YES. Allow this member to coordinate
 parallel processing on other members.
 6 ASSISTANT ====> NO NO or YES. Allow this member to assist
 with parallel processing.

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 29. Define Group or Member Panel: DSNTIPK

### 1. GROUP NAME

Acceptable values: 1-8 characters made up of A-Z, 0-9, \$, #, @  
Default: DSNCAT  
DSNZPxxx: DSN6GRP GRPNAME

Specify the name of a new or existing DB2 data sharing group. The group name encompasses the entire data sharing group and forms the basis for the coupling facility structure names.

To avoid names that IBM uses for its MVS cross-system coupling facility (XCF) groups, the first character must be an upper-case letter J-Z unless the name begins with DSN. Do not use SYS as the first three characters, and do not use UNDESIG as the group name.

### 2. MEMBER NAME

Acceptable values: 1-8 characters  
Default: DSN1  
DSNZPxxx: DSN6GRP MEMBNAME

Specify the name of a new or existing DB2 data sharing member. We recommend that you use the MVS subsystem name. DB2 uses this name as its XCF member

name. An example of a member name is DB1G. The member name can consist of the characters A-Z, 0-9, \$, #, and @.

### 3. WORK FILE DB

|                    |                |
|--------------------|----------------|
| Acceptable values: | 1-8 characters |
| Default:           | DSN1           |
| DSNZPxxx:          | none           |

Specify the name of the work file database for the DB2 member. Each DB2 member has its own work file database (called DSNDB07 in a non-data-sharing environment). One member of the data sharing group can have the name DSNDB07, but we recommend that you use a more meaningful name, such as WRKDSN1. You cannot specify a name that begins with DSNDB unless the name is DSNDB07.

### 4. GROUP ATTACH

|                    |                |
|--------------------|----------------|
| Acceptable values: | 1-4 characters |
| Default:           | none           |
| DSNZPxxx:          | none           |

Specify a generic group attachment name for batch programs, the call attachment facility (CAF), and utilities. An example of a group attachment name is DBOG. See *Data Sharing: Planning and Administration* for information about using the group attachment name.

### 5. COORDINATOR

|                    |                  |
|--------------------|------------------|
| Acceptable values: | YES, NO          |
| Default:           | NO               |
| DSNZPxxx:          | DSN6GRP COORDNTR |

Specify whether this DB2 member can coordinate parallel processing on other members of the group. If you specify NO, then a query can be processed by this DB2 member only. If you specify YES, then a read-only query running on this DB2 member can be processed in part on other members of the group.

### 6. ASSISTANT

|                    |                |
|--------------------|----------------|
| Acceptable values: | YES, NO        |
| Default:           | NO             |
| DSNZPxxx:          | DSN6GRP ASSIST |

Specify whether this DB2 member can assist a parallelism coordinator with parallel processing. If you specify NO, this member is not considered as an assistant at either bind time or run time. If you specify YES, this member is considered at both bind time and run time. To qualify as an assistant at run time, the VPPSEQT and VPXPSEQT buffer pool thresholds of this member must each be greater than zero.

## System Resource Data Set Names: DSNTIPH

The entries on this panel name the bootstrap data sets, active logs, and archive logs. They also specify the number of copies (single or dual logging) for the active and archive logs.

Fields 1, 2, 4, 5, 7, and 8 on the DSNTIPH panel contain the prefix that was entered in the CATALOG ALIAS field on installation panel DSNTIPA2. If you scroll back to panel DSNTIPA2 and change the CATALOG ALIAS value, the values for fields 1, 2, 4, 5, 7, and 8 change. When you scroll from panel DSNTIPA2 to panel DSNTIPH, check these values and enter them again if necessary. This is not a concern in MIGRATE or UPDATE modes because the CATALOG ALIAS value cannot be changed.

# Dual logging improves reliability of recovery and, for active log reads, eases device  
# contention. We strongly recommend that you specify dual logging for both active  
and archive logs. If you do, and an error occurs during offload to the archive logs,  
DB2 restarts the archive process using the second copy of the active log. If you  
use dual active logging, it is strongly advised that you place the two active logs on  
different DASD volumes and, ideally, on different channels and control units. To do  
that, specify different volume serial numbers for VOLUME SERIAL 6 and VOLUME  
SERIAL 7 (fields 8 and 9 on panel DSNTIPA2).

**If you are migrating**, DB2 for OS/390 Version 5 adopts your DB2 for MVS/ESA Version 4 BSDS and active logs. Therefore, you cannot change the parameters that affect the characteristics of these objects during migration. However, after migration, you can update the parameters that affect the BSDS and active logs.

```
DSNTIPH INSTALL DB2 - SYSTEM RESOURCE DATA SET NAMES
===>
DSNT443I Values marked with an asterisk have been updated
Enter data below:

Bootstrap Data Sets (BSDS):

* 1 COPY 1 NAME ===> DSNCAT.BSDS01
* 2 COPY 2 NAME ===> DSNCAT.BSDS02

Active Logs:
 3 NUMBER OF COPIES ===> 2 2 or 1. Number of active log copies
* 4 COPY 1 PREFIX ===> DSNCAT.LOGCOPY1
* 5 COPY 2 PREFIX ===> DSNCAT.LOGCOPY2

Archive Logs:
 6 NUMBER OF COPIES ===> 2 2 or 1. Number of archive log copies
* 7 COPY 1 PREFIX ===> DSNCAT.ARCHLOG1
* 8 COPY 2 PREFIX ===> DSNCAT.ARCHLOG2
 9 TIMESTAMP ARCHIVES ===> NO YES or NO

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 30. System Resource Data Set Names: DSNTIPH



## 1. COPY 1 NAME

|                    |                                                   |
|--------------------|---------------------------------------------------|
| Acceptable values: | valid data set name; 1-33 characters              |
| Default:           | DSNCAT.BSDS01 or<br>DSNCAT.DSN1.BSDS01            |
| Update:            | see Update on page 2-228; not during<br>migration |
| DSNZPxxx:          | none                                              |

Specify the fully qualified name of the first copy of the bootstrap data set. For non-data-sharing environments, the default prefix is DSNCAT.BSDSxx. For data sharing environments, the default prefix is DSNCAT.DSN1.BSDSxx. The resulting data set name is DSNCAT.BSDSxx.Annnnnnn or DSNCAT.DSN1.BSDSx.Annnnnnn where:

- DSNCAT is the value you specified for CATALOG ALIAS (field 1 on installation panel DSNTIPA2). You can change this portion of the data set prefix on this panel. If you change it, then another catalog alias is needed. This additional catalog alias will not be automatically defined by the installation process.
- DSN1 is the value you specified for MEMBER NAME on panel DSNTIPK.
- xx is **01** for the first copy of the logs and **02** for the second copy.
- *Annnnnnn* is generated internally.

For the definition of a valid data set name, see page 2-101.

## 2. COPY 2 NAME

|                    |                                                   |
|--------------------|---------------------------------------------------|
| Acceptable values: | valid data set name; 1-33 characters              |
| Default:           | DSNCAT.BSDS02 or<br>DSNCAT.DSN1.BSDS02            |
| Update:            | see Update on page 2-228; not during<br>migration |
| DSNZPxxx:          | none                                              |

Specify the fully qualified name of the second copy of the bootstrap data set. For the definition of a valid data set name, see page 2-101.

## 3. NUMBER OF COPIES

|                    |                                                   |
|--------------------|---------------------------------------------------|
| Acceptable values: | 1-2                                               |
| Default:           | 2                                                 |
| Update:            | see Update on page 2-228; not during<br>migration |
| DSNZPxxx:          | DSN6LOGP TWOACTV                                  |

Specify the number of copies of the active log to be maintained: 2 (dual logging) or 1 (single logging). Dual logging increases reliability of recovery. If your DB2 subsystem creates copies of the archive log on tape, two tape drives must be available during the off-load process.

## 4. COPY 1 PREFIX

|                    |                                                   |
|--------------------|---------------------------------------------------|
| Acceptable values: | valid data set name prefix; 1-30 characters       |
| Default:           | DSNCAT.LOGCOPY1 or<br>DSNCAT.DSN1.LOGCOPY1        |
| Update:            | see Update on page 2-228; not during<br>migration |
| DSNZPxxx:          | none                                              |

Specify the prefix for the first copy of the active log data sets. For non-data-sharing environments, the default prefix is DSNCAT.LOGCOPYx. For data sharing environments, the default prefix is DSNCAT.DSN1.LOGCOPYx. The resulting data set name is DSNCAT.LOGCOPYx.Annnnnnnn or DSNCAT.DSN1.LOGCOPYx.Annnnnnnn where:

- DSNCAT is the value you specified for CATALOG ALIAS (field 1 on installation panel DSNTIPA2). You can change this portion of the data set prefix on this panel. If you change it, then another catalog alias is needed. This additional catalog alias will not be automatically defined by the installation process.
- DSN1 is the value you specified for MEMBER NAME on panel DSNTIPK.
- LOGCOPY is part of the data set prefix that you can change on this panel.
- x is **1** for the first copy of the logs and **2** for the second copy.
- nn is the data set number.

For information on valid data set names, see page 2-101.

## 5. COPY 2 PREFIX

|                    |                                                   |
|--------------------|---------------------------------------------------|
| Acceptable values: | valid data set name prefix; 1-30 characters       |
| Default:           | DSNCAT.LOGCOPY2 or<br>DSNCAT.DSN1.LOGCOPY2        |
| Update:            | see Update on page 2-228; not during<br>migration |
| DSNZPxxx:          | none                                              |

Specify the prefix for the second copy of the active log data sets. See the description in field 4. If you are using single logging, accept the default value. Do not leave the entry blank.

## 6. NUMBER OF COPIES

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | 1-2                       |
| Default:           | 2                         |
| Update:            | option 3 on panel DSNTIPB |
| DSNZPxxx:          | DSN6LOGP TWOARCH          |

Specify the number of copies of the archive log to be produced during off-loading: 2 (dual logging) or 1 (single logging). Dual logging increases reliability of recovery.

## 7. COPY 1 PREFIX

|                    |                                             |
|--------------------|---------------------------------------------|
| Acceptable values: | valid data set name prefix; 1-35 characters |
| Default:           | DSNCAT.ARCHLOG1 or<br>DSNCAT.DSN1.ARCLG1    |
| Update:            | option 3 on panel DSNTIPB                   |
| DSNZPxxx:          | DSN6ARVP ARCPFX1                            |

Specify the prefixes of the first and second copies of archive log data sets. For definitions of valid data set names, see page 2-101.

## 8. COPY 2 PREFIX

|                    |                                             |
|--------------------|---------------------------------------------|
| Acceptable values: | valid data set name prefix; 1-35 characters |
| Default:           | DSNCAT.ARCHLOG2 or<br>DSNCAT.DSN1.ARCLG2    |
| Update:            | option 3 on panel DSNTIPB                   |
| DSNZPxxx:          | DSN6ARVP ARCPFX2                            |

See the description of field 7. If you are using single logging, accept the default value. Do not leave the entry blank.

## 9. TIMESTAMP ARCHIVES

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | YES, NO                   |
| Default:           | NO                        |
| Update:            | option 3 on panel DSNTIPB |
| DSNZPxxx:          | DSN6ARVP TSTAMP           |

Specify whether or not the date and time of creation of the DB2 archive log data set is placed in the archive log data set name.

If you specify YES, the maximum allowable length of the user-controlled portion of the archive log prefix is reduced from 35 characters to 19 characters. This reduction in size permits the 16-character date and time qualifiers (timestamp) to be added to the archive log data set prefix. The timestamp format is as follows:

*.Dyyddd.Thhmmss.t,*

where:

|            |                                     |
|------------|-------------------------------------|
| <i>D</i>   | is the letter D                     |
| <i>yy</i>  | are the last two digits of the year |
| <i>ddd</i> | is the day of the year              |
| <i>T</i>   | is the letter T                     |
| <i>hh</i>  | is the hour                         |
| <i>mm</i>  | are the minutes                     |
| <i>ss</i>  | are the seconds                     |
| <i>t</i>   | are tenths of a second              |

---

## Data Set Names Panel 1: DSNTIPT

The entries on this panel establish data set names for the DB2 libraries used in the DB2 CLIST and JCL provided by DB2. The values entered on this panel are edited into all pertinent sample and installation jobs.

You can fill in these values one of three ways: same data set name prefix, no data set name prefix, or a new data set name prefix. Table 47 summarizes these selections.

*Table 47. Summary of Values*

| <b>If You Use ...</b>                       | <b>Then</b>                                                                   |
|---------------------------------------------|-------------------------------------------------------------------------------|
| Same data set name prefix or data set names | Current data sets are deleted and reallocated for installation and migration. |
| No data set names                           | No new output is created. Previous output remains intact.                     |
| New prefix                                  | Output saved in new data set. Previous output remains intact.                 |

The following warning message is displayed for any output data set that already exists:

DSNT434I WARNING, DATA SETS MARKED WITH ASTERISKS EXIST AND WILL BE OVERWRITTEN

In order to avoid deleting these data sets, do one of the following:

- Press RETURN to leave the installation process.
- Change the data set names.

Press ENTER again if you want to continue; this overwrites your data sets.

When you are in update mode, this panel is displayed immediately after panel DSNTIPA1. This allows you to check the SDSNSAMP data set name to see if it is the one you want to use for the DSNTIJUZ job. Data sets are not deleted or reallocated if you use the same name. Instead, the data set is compressed, and only the DSNTIJUZ member is replaced within the data set. Other members in the data set are left unchanged.

```

DSNTIPT INSTALL DB2 - DATA SET NAMES PANEL 1
===> _

Data sets allocated by the installation CLIST for edited output:
 1 TEMP CLIST LIBRARY ===> prefix.NEW.SDSNTEMP
 2 SAMPLE LIBRARY ===> prefix.NEW.SDSNSAMP

Data sets allocated by the installation jobs:
 3 CLIST LIBRARY ===> prefix.NEW.SDSNCLST
 4 APPLICATION DBRM ===> prefix.DBRMLIB.DATA.suffix
 5 APPLICATION LOAD ===> prefix.RUNLIB.LOAD.suffix
 6 DECLARATION LIBRARY===> prefix.SRCLIB.DATA.suffix

Data sets allocated by SMP/E and other methods:
 7 LINK LIST LIBRARY ===> prefix.SDSNLINK.suffix
 8 LOAD LIBRARY ===> prefix.SDSNLOAD.suffix
 9 MACRO LIBRARY ===> prefix.SDSNMACS.suffix
10 LOAD DISTRIBUTION ===> prefix.ADSNLOAD.suffix
11 EXIT LIBRARY ===> prefix.SDSNEXIT.suffix
12 DBRM LIBRARY ===> prefix.SDSNDBRM.suffix
13 IRLM LOAD LIBRARY ===> prefix.SDXRRESL.suffix

PRESS: ENTER to continue RETURN to exit HELP for more information

```

Figure 31. Data Set Names Panel 1: DSNTIPT

**Fields 1 and 2** are data sets allocated by the installation CLIST for edited output. Field 3 is allocated by DSNTIJVC. If the input member is DSNTIDXA (field 6 on installation panel DSNTIPA1 on page 2-99), the three data sets default to *prefix.NEW.SDSNTEMP*, *prefix.NEW.SDSNCLST*, and *prefix.NEW.SDSNSAMP* respectively, where *prefix* is the value entered for field 4 on installation panel DSNTIPA1.

Table 48. Job Tailoring Fields

| Mode       | Tailored output          | No tailored output     |
|------------|--------------------------|------------------------|
| Installing | All three fields entered | All three fields blank |
| Migrating  | All three fields entered | All three fields blank |
| Updating   | SAMPLE LIBRARY entered   | SAMPLE LIBRARY blank   |

These fields are blanked after a successful tailoring session to avoid writing over the tailored output.

**1. TEMP CLIST LIBRARY**

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix.NEW.SDSNTEMP</i>          |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name where edited CLISTs are placed. This field must not be blank if tailoring is wanted.

## 2. SAMPLE LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix</i> .NEW.SDSNSAMP         |
| Update:            | option 4 on panel DSNTIPB           |
| DSNZPxxx:          | none                                |

Specify the name of the edited JCL library. In Update mode, the new sample library data set is not reallocated. It is compressed and member DSNTIJUZ is overwritten. This field must not be blank if tailoring is wanted.

## 3. CLIST LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix</i> .NEW.SDSNCLST         |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name that job DSNTIJVC loads all CLISTs into. This field must not be blank if tailoring is wanted.

**Fields 4, 5, and 6** are for DB2 provided sample applications. The names of your own development libraries most likely are different from the defaults shown here. Job DSNTIJMV references another set of DBRMLIB, RUNLIB, and SRCLIB data sets for SYS1.PROCLIB. See "Installation Step 1: Define DB2 to MVS: DSNTIJMV" on page 2-233 for more information. These fields must not be blank.

## 4. APPLICATION DBRM

|                    |                                            |
|--------------------|--------------------------------------------|
| Acceptable values: | valid data set name: see page 2-101        |
| Default:           | <i>prefix</i> .DBRMLIB.DATA. <i>suffix</i> |
| Update:            | cannot change during update                |
| DSNZPxxx:          | none                                       |

Specify the name of the library for DB2 sample application DBRMs.

## 5. APPLICATION LOAD

|                    |                                           |
|--------------------|-------------------------------------------|
| Acceptable values: | valid data set name: see page 2-101       |
| Default:           | <i>prefix</i> .RUNLIB.LOAD. <i>suffix</i> |
| Update:            | cannot change during update               |
| DSNZPxxx:          | none                                      |

Specify the name of the DB2 sample application load module library.

## 6. DECLARATION LIBRARY

|                    |                                           |
|--------------------|-------------------------------------------|
| Acceptable values: | valid data set name: see page 2-101       |
| Default:           | <i>prefix</i> .SRCLIB.DATA. <i>suffix</i> |
| Update:            | cannot change during update               |
| DSNZPxxx:          | none                                      |

Specify the name of the DB2 declaration library for sample application include files.

**Fields 7 through 13** specify the names of data sets allocated during SMP processing. These fields must not be blank.

## 7. LINK LIST LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix.SDSNLINK.suffix</i>       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the name of the APF-authorized DB2 early code library.

## 8. LOAD LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix.SDSNLOAD.suffix</i>       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the name of the main APF-authorized DB2 load module library.

## 9. MACRO LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix.SDSNMACS.suffix</i>       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the name of the library that contains the CICS and IMS attachment facility macros, the initialization parameter macros, and some data mapping macros needed for some applications.

## 10. LOAD DISTRIBUTION

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix.ADSNLOAD.suffix</i>       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the name of the distribution load module library.

## 11. EXIT LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix.SDSNEXIT.suffix</i>       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the name of the library where your DSNZPxxx module, DSNHDECP module, and exit routines are placed. When *prefix.SDSNLOAD* and *prefix.SDSNEXIT* are used together, list *prefix.SDSNEXIT* first to override the IBM defaults in *prefix.SDSNLOAD*.

## 12. DBRM LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix.SDSNDBRM.suffix</i>       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the name of the library where your SPUFI, DCLGEN, flagger, and schema processor DBRMs are placed.

### 13. IRLM LOAD LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | <i>prefix.SDXRRESL.suffix</i>       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the name of the IRLM load library data set to use in the IRLM procedure.



## Data Set Names Panel 2: DSNTIPU

The entries on this panel and the following panels, DSNTIPQ and DSNTIPG, establish data set names for other product libraries. The values entered on these panels are edited into sample and installation jobs. If you do not have the product, accept the default. Jobs for those particular products should not be run.

DB2 makes assumptions about which one of the possible C or C++ compilers you are using depending on the values you supply or leave as default in the C or C++ fields.

Many data set names for other products appear in the jobs. Most of these data sets can be entered on this panel and on installation panel DSNTIPW. These names are shown in Table 49 as they appear in the jobs shipped with DB2. Change the names of the data sets if they are different at your site.

Table 49 (Page 1 of 2). Data Set Names Used in Jobs for Products

| Job      | Data Set Name                                                                  | Function                                                                                                                                                                                               |
|----------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DSNTEJ1  | SYS1.MACLIB<br>SYS1.SORTLIB                                                    | Assembler macro library<br>DFSORT load modules (can be deleted if DFSORT is in link list)                                                                                                              |
| DSNTEJ1P | SYS1.PLILINK<br>SYS1.SIBMLINK                                                  | PL/I V2R3 dynamic runtime base library<br>PL/I V2R3 dynamic runtime common library                                                                                                                     |
| DSNTEJ2A | SYS1.SORTLIB                                                                   | DFSORT load modules (can be deleted if DFSORT is in link list)                                                                                                                                         |
| DSNTEJ2C | CEE.V1R5M0.SCEERUN                                                             | Language Environment Version 1 Release 5 dynamic runtime library                                                                                                                                       |
| DSNTEJ2D | CEE.V1R4M0.SCEERUN                                                             | Language Environment Version 1 Release 4 dynamic runtime library                                                                                                                                       |
| DSNTEJ2E | CEE.V1R4M0.SCEERUN                                                             | Language Environment Version 1 Release 4 dynamic runtime library                                                                                                                                       |
| DSNTEJ2F | SYS1.MACLIB<br>SYS1.VSF2FORT                                                   | Assembler macro library<br>VS FORTRAN runtime library                                                                                                                                                  |
| DSNTEJ2P | SYS1.PLILINK<br>SYS1.SIBMLINK                                                  | PL/I V2R3 dynamic runtime base library<br>PL/I V2R3 dynamic runtime common library                                                                                                                     |
| DSNTEJ3C | CEE.V1R5M0.SCEERUN                                                             | Language Environment Version 1 Release 5 dynamic runtime library                                                                                                                                       |
| DSNTEJ3P | SYS1.PLILINK<br>SYS1.SIBMLINK                                                  | PL/I V2R3 dynamic runtime base library<br>PL/I V2R3 dynamic runtime common library                                                                                                                     |
| DSNTEJ4C | IMSVS.RESLIB<br>CEE.V1R5M0.SCEERUN                                             | IMS linkage editor library<br>Language Environment Version 1 Release 5 dynamic runtime library                                                                                                         |
| DSNTEJ4P | IMSVS.RESLIB<br>SYS1.PLIBASE<br>SYS1.PLILINK<br>SYS1.SIBMBASE<br>SYS1.SIBMLINK | IMS linkage editor library<br>PL/I V2R3 linkage editor base library<br>PL/I V2R3 dynamic runtime base library<br>PL/I V2R3 linkage editor common library<br>PL/I V2R3 dynamic runtime common library   |
| DSNTEJ5A | CICS330.SDFHLOAD<br>CICS330.SDFHMAC<br>SYS1.MACLIB                             | CICS command translator and linkage editor<br>CICS macro library<br>Assembler macro library                                                                                                            |
| DSNTEJ5C | CICS330.SDFHLOAD<br>IGY.V1R2M0.SIGYCOMP(IGYCRCTL)                              | CICS command translator and linkage editor library<br>IBM COBOL for MVS & VM (V1R2) compiler load module<br>See also the list of libraries used by DSNH CLIST in Chapter 2 of <i>Command Reference</i> |
| DSNTEJ5P | CICS330.SDFHLOAD<br>CICS330.SDFHPLI<br>SYS1.PLIBASE<br>SYS1.SIBMBASE           | CICS command translator and linkage editor library<br>CICS PL/I linkage editor library<br>PL/I V2R3 linkage editor base library<br>PL/I V2R3 linkage editor common library                             |
| DSNTEJ6P | CEE.V1R5M0.SCEERUN                                                             | Language Environment Version 1 Release 5 dynamic runtime library                                                                                                                                       |
| DSNTEJ6D | CEE.V1R5M0.SCEERUN                                                             | Language Environment Version 1 Release 5 dynamic runtime library                                                                                                                                       |

Table 49 (Page 2 of 2). Data Set Names Used in Jobs for Products

| Job                     | Data Set Name                   | Function                                                         |
|-------------------------|---------------------------------|------------------------------------------------------------------|
| DSNTEJ6S                | CEE.V1R5M0.SCEERUN              | Language Environment Version 1 Release 5 dynamic runtime library |
| DSNTEJ6T                | CEE.V1R5M0.SCEERUN              | Language Environment Version 1 Release 5 dynamic runtime library |
| # DSNTEJ61              | CEE.V1R5M0.SCEERUN              | Language Environment Version 1 Release 5 dynamic runtime library |
| # DSNTEJ62              | CEE.V1R5M0.SCEERUN              | Language Environment Version 1 Release 5 dynamic runtime library |
| # DSNTIJMV <sup>1</sup> | SYS1.MACLIB                     | Assembler macro library                                          |
|                         | EDC.V1R2M0.SEDCDCMP             | C compiler library                                               |
|                         | EDCDC120                        | C compiler load module                                           |
|                         | EDC.V1R2M0.SEDCDMSG(EDCMSGE)    | C compiler message file                                          |
|                         | CEE.V1R4M0.SCEERUN(EDCPRLK)     | C compiler prelink load module                                   |
|                         | CEE.V1R4M0.SCEELKED             | Language Environment Version 1 Release 4 linkage editor library  |
|                         | CEE.V1R4M0.SCEEMSGP(EDCPMSGE)   | C prelink message file                                           |
|                         | CEE.V1R4M0.SCEECPP              | C++autolink library                                              |
|                         | CBC.V3R1M1.SCBC3CPP             | C++compiler messages                                             |
|                         | CBC.V3R1M1.SCLB3CPP             | C++class library                                                 |
|                         | CEE.V1R4M0.SCEERUN              | Language Environment Version 1 Release 4 dynamic runtime library |
|                         | CBC.V3R1M1.SCLB3H.H             | C++library headers                                               |
|                         | CEE.V1R4M0.SCEEH.H              | C++C library headers                                             |
|                         | CEE.V1R4M0.SCEEMSGP             | Language Environment Version 1 Release 4 prelink editor messages |
|                         | CICS330.SDFHCOB                 | CICS library for OS/VS COBOL                                     |
|                         | SYS1.COB2CICS                   | CICS COBOL II library                                            |
|                         | CICS330.SDFHLOAD                | CICS command translator and linkage editor                       |
|                         | <i>prefix</i> .SDSNLOAD(DSNHPC) | DB2 precompiler                                                  |
|                         | <i>prefix</i> .SDSNLOAD         | DB2 linkage editor library                                       |
|                         | SYS1.COBLIB                     | OS/VS COBOL linkage editor library                               |
|                         | IKFCBL00                        | OS/VS COBOL compiler load module                                 |
|                         | SYS1.V1R4.COB2COMP              | COBOL II compiler load module                                    |
|                         | SYS1.V1R4.COB2LIB               | COBOL II linkage editor library                                  |
|                         | IGY.V1R2M0.SIGYCOMP(IGYCRCTL)   | IBM COBOL for MVS & VM (V1R2) compiler load module               |
|                         | CEE.V1R5M0.SCEELKED             | Language Environment Version 1 Release 5 linkage editor library  |
|                         | CEE.V1R5M0.SCEEMSGP             | Language Environment Version 1 Release 5 prelink editor messages |
|                         | CEE.V1R5M0.SCEERUN              | Language Environment Version 1 Release 5 runtime library         |
|                         | SOMMVS.V1R1M0.SGOSPLKD          | SOMobjects for MVS library                                       |
|                         | IMSVS.RESLIB                    | IMS linkage editor library                                       |
|                         | ISP.V3R5M0.ISPLOAD              | ISPF ISPLINK module                                              |
|                         | IEL0AA                          | PL/I V2R3 compiler load module                                   |
|                         | SYS1.PLILINK                    | PL/I V2R3 dynamic runtime base library                           |
|                         | SYS1.SIBMLINK                   | PL/I V2R3 dynamic runtime common library                         |
|                         | SYS1.PLIBASE                    | PL/I V2R3 linkage editor base library                            |
|                         | SYS1.SIBMBASE                   | PL/I V2R3 linkage editor common library                          |
|                         | IEL.V1R1M1.SIELCOMP(IEL1AA)     | PL/I for MVS & VM (V1R1.1)                                       |

<sup>1</sup>. Language Environment Version 1 Release 4 is needed for C/C++ V3 R1. Language Environment Version 1 Release 5 is needed for IBM COBOL for MVS & VM (V1R2) and PL/I for MVS & VM (V1R1.1)

When the compiler fields are left blank, the DSNH CLIST and the provided JCL procedures operate differently. The DSNH CLIST issues a specific call statement, using the default load module data set name as the argument of the call. The JCL procedures use the MVS link list to find the data set in which the load module resides.

```

DSNTIPU INSTALL DB2 - DATA SET NAMES PANEL 2
===>

Enter data set names below:
 1 C COMPILER ===> EDC.V1R2M0.SEDCDCMP
 2 C COMPILER MESSAGES ===> EDC.V1R2M0.SEDCDMSG
 3 C PRE-LINK MESSAGES ===> CEE.V1R4M0.SCEEMSGP
 4 C LIBRARY HEADERS ===> EDC.V1R2M0.SEDCDHDR
 5 C LINK EDIT STUBS ===> CEE.V1R4M0.SCEELKED
 6 C DYNAMIC RUNTIME ===> CEE.V1R4M0.SCEERUN
 7 C PROGRAM NAME ===> EDCDC120
 8 CPP COMPILER ===>
 9 CPP COMPILER MESSAGES ===> CBC.V3R1M1.SCBC3MSG
10 CPP PRE-LINK MESSAGES ===> CEE.V1R4M0.SCEEMSGP
11 CPP LINK EDIT STUBS ===> CEE.V1R4M0.SCEELKED
12 CPP DYNAMIC RUNTIME ===> CEE.V1R4M0.SCEERUN
13 CPP PROGRAM NAME ===> CBC310PP
14 CPP STANDARD HEADERS ===> CEE.V1R4M0.SCEEH.H
15 CPP CLASS LIB HEADERS ===> CBC.V3R1M1.SCLB3H.H
16 CPP AUTO CALL LIBRARY ===> CEE.V1R4M0.SCEECPP
17 CPP CLASS LIBRARY ===> CBC.V3R1M1.SCLB3CPP

PRESS: ENTER to continue RETURN to exit HELP for more information

```

Figure 32. Data Set Names Panel 2: DSNTIPU

### 1. C COMPILER

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | EDC.V1R2M0.SEDCDCMP                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C compiler library. If you specify a name in this field, a STEPLIB is added to each job provided by DB2 that uses this compiler.

These C fields can include AD/Cycle C/370 V1R2 (C only), C/C++ for MVS/ESA V3R1, or C/C++ for MVS/ESA V3R2. C/C++ for MVS/ESA V3R1 and V3R2 are shipped as single products, but it is necessary to define them separately on this line and on line 8 depending on your need for C or C++. If C is not installed, skip jobs DSNTJE2D, DSNTJE6D, and DSNTJE6T and remove both the DSNHC procedure from job DSNTIJMV and the statement that renames the procedure from the DSNTIJFV job.

### 2. C COMPILER MESSAGES

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | EDC.V1R2M0.SEDCDMSG                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C compiler message file. If you specify C/C++ for MVS/ESA V3R2 on line 1, you do not need to specify anything on this line.

### 3. C PRE-LINK MESSAGES

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R4M0.SCEEMSGP                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name for the C pre-link message file. If you specify this file name, DB2 assumes you are using SAA C/370. Do not specify this value if you are using the IBM C/370 compiler.

### 4. C LIBRARY HEADERS

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | EDC.V1R2M0.SEDCDHDR                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C header files.

### 5. C LINK EDIT STUBS

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R4M0.SCEELKED                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C linkage editor library.

### 6. C DYNAMIC RUNTIME

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R4M0.SCEERUN                            |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C runtime library.

### 7. C PROGRAM NAME

|                    |                                                  |
|--------------------|--------------------------------------------------|
| Acceptable values: | blank, or valid load module name: see page 2-101 |
| Default:           | EDCDC120                                         |
| Update:            | cannot change during update                      |
| DSNZPxxx:          | none                                             |

Specify the C program name.

### 8. CPP COMPILER

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | none                                          |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C++ compiler library.

You can specify C/C++ for MVS/ESA V3R1, or C/C++ for MVS/ESA V3R2. These are shipped as single products, but it is necessary to define them separately on this line and on line 1 depending on your need for C or C++.

## 9. CPP COMPILER MESSAGES

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CBC.V3R1M1.SCBC3MSG                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name for C++ compiler messages file. If you specify a name in this field, a STEPLIB is added to each job provided by DB2 that uses this compiler. If C++ is not installed, skip sample job DSNTJ2E. If you specify C/C++ for MVS/ESA V3R2 on line 8, you do not need to specify anything on this line.

## 10. CPP PRE-LINK MESSAGES

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R4M0.SCEEMSGP                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name for C++ pre-link messages file.

## 11. CPP LINK EDIT STUBS

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R4M0.SCEELKED                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C++ linkage editor library.

## 12. CPP DYNAMIC RUNTIME

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R4M0.SCEERUN                            |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name for the C++ runtime library.

### 13. CPP PROGRAM NAME

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CBC310PP                                      |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the C++ program name.

### 14. CPP STANDARD HEADERS

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R4M0.SCEEH.H                            |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C++ standard header files.

### 15. CPP CLASS LIB HEADERS

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CBC.V3R1M1.SCLB3H.H                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C++ class library header files.

### 16. CPP AUTO CALL LIBRARY

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R4M0.SCEECPP                            |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C++ auto call library.

### 17. CPP CLASS LIBRARY

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CBC.V3R1M1.SCLB3CPP                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the C++ class library.

## Data Set Names Panel 3: DSNTIPQ

The entries on this panel establish data set names for VS COBOL and IBM COBOL.

```
DSNTIPQ INSTALL DB2 - DATA SET NAMES PANEL 3
====>

Enter data set names below:
 1 VS COBOL COMPILER ==>
 2 VS COBOL LINK EDIT ==> SYS1.COBLIB
 3 VS COBOL II COMPILER ==>
 4 VS COBOL II LINK EDIT ==> SYS1.V1R4.COB2LIB
 5 IBM COBOL COMPILER ==>
 6 IBM COBOL RUNTIME ==> CEE.V1R5M0.SCEERUN
 7 IBM COBOL PRE-LINK MSGS ==> CEE.V1R5M0.SCEEMSGP
 8 IBM COBOL LINK EDIT ==> CEE.V1R5M0.SCEELKED
 9 IBM COBOL SOM LIBRARY ==> SOMMVS.V1R1M0.SGOSPLKD
10 SOM DLL IMPORT LIBRARY ==> SOMMVS.V1R1M0.SGOSIMP

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 33. Data Set Names Panel 3: DSNTIPQ

If COBOL is not installed on your system, do not run jobs DSNTJ2C, DSNTJ3C, DSNTJ4C, DSNTJ5C, or DSNTJ6. The current default for DB2 is IBM COBOL for MVS and VM.

### 1. VS COBOL COMPILER

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | none                                          |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the COBOL compiler library. If you specify a name in this field, a STEPLIB is added to each DB2 job that uses this compiler.

If you do not have OS/VS COBOL at your site, remove the DSNHCOB procedure from job DSNTIJMV and the statement that renames the procedure from job DSNTIJFV. See “Special Considerations for COBOL Programs” on page 2-308 for the specific changes.

### 2. VS COBOL LINK EDIT

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.COBLIB                         |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the COBOL linkage library.

### 3. VS COBOL II COMPILER

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | none                                          |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the COBOL II compiler library. If you specify a name in this field, a STEPLIB is added to each DB2 job that uses this compiler. If you have VS COBOL II, you can use the COBOL samples.

If you do not have VS COBOL II at your site, remove the DSNHCOB2 procedure from job DSNTIJMV and the statement that renames the procedure from jobs DSNTIJFV and DSNTIJMV.

### 4. VS COBOL II LINK EDIT

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.V1R4.COB2LIB                   |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the COBOL II linkage editor library.

### 5. IBM COBOL COMPILER

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | none                                          |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the COBOL compiler load module library.

If you do not have IBM COBOL at your site, remove the DSNHICOB procedure from job DSNTIJMV and the statement that renames the procedure from jobs DSNTIJFV and DSNTIJMV.

### 6. IBM COBOL RUNTIME

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R5M0.SCEERUN                            |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the COBOL runtime library.

### 7. IBM COBOL PRELINK MSGS

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R5M0.SCEEMSGP                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the COBOL pre-link messages.



## 8. IBM COBOL LINK EDIT

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | CEE.V1R5M0.SCEELKED                           |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the COBOL link edit library.

## 9. IBM COBOL SOM LIBRARY

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | SOMMVS.V1R1M0.SGOSPLKD                        |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the COBOL SOM library. Blank out the value for IBM COBOL SOM LIBRARY, if you are using OS/390 Version 1 Release 3 or higher.

## 10. SOM DLL IMPORT LIBRARY

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | SOMMVS.V1R1M0.SGOSIMP                         |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the SOM DLL import library. This data set name is valid only with IBM COBOL for OS/390 & VM Version 2 Release 1.

To prepare application programs to link to the SOM/MVS kernal and classes from the DSOM object library, blank out the value in this field. This action is valid with both IBM COBOL for MVS & VM Version 1 Release 2 and IBM COBOL for OS/390 & VM Version 2 Release 1 only if your operating system precedes OS/390 Version 1 Release 3.

To prepare non-object-oriented application programs, blank out the values for IBM COBOL SOM LIBRARY and SOM DLL IMPORT LIBRARY fields. This is valid for both IBM COBOL for MVS & VM Version 1 Release 2 and IBM COBOL for OS/390 & VM Version 2 Release 1.

## Data Set Names Panel 4: DSNTIPG

The entries on this panel establish data set names for other product libraries. This panel is an extension to panel DSNTIPU—see the full description in “Data Set Names Panel 2: DSNTIPU” on page 2-121.

DB2 makes assumptions about which one of three possible PL/I compilers you are using depending on the values you supply or leave as default in the PL/I fields. If you specify values in all the PL/I fields, DB2 assumes you are using the OS PL/I Version 2 compiler. If you do not specify values for both the PL/I LINK EDIT COMMON and PL/I DYNAMIC RUNTIME COMMON fields, then DB2 assumes you are using the PL/I MVS and VM compiler. This compiler is required for running stored procedures sample programs. If you do not specify values for the PL/I LINK EDIT COMMON, PL/I DYNAMIC RUNTIME BASE, and PL/I DYNAMIC RUNTIME COMMON fields, then DB2 assumes you are using the OS PL/I Version 1 compiler.

```
DSNTIPG INSTALL DB2 - DATA SET NAMES PANEL 4
====>

Enter data set names below:
 1 ASSEMBLER ====>
 2 FORTRAN COMPILER ====>
 3 FORTRAN LINK EDIT ====> SYS1.VSF2FORT
 4 PL/I COMPILER ====>
 5 PL/I LINK EDIT BASE ====> SYS1.PLIBASE
 6 PL/I LINK EDIT COMMON ====> SYS1.SIBMBASE
 7 PL/I DYN RUNTIME BASE ====> SYS1.PLILINK
 8 PL/I DYN RUNTIME COMMON ====> SYS1.SIBMLINK
 9 CPP PROCEDURE LIBRARY ====> CBC.V3R1M1.SCBC3UTL
10 LE/370 RUNTIME LIBRARY ====> CEE.V1R5M0.SCEERUN

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 34. Data Set Names Panel 4: DSNTIPG

### 1. ASSEMBLER

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | none                                          |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the assembler load module library. If you specify a value for this field, a STEPLIB is added to each job provided by DB2 that uses the assembler. This field can be left blank if the library is in the link list.

## 2. FORTRAN COMPILER

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | none                                          |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the FORTRAN compiler invocation load module library. If you specify a name in this field, a STEPLIB is added to each job provided by DB2 that uses this compiler.

If VS FORTRAN is not installed on your system, do not run job DSNTEJ2F. Remove the DSNHFOR procedure from job DSNTIJMV and the statement that renames the procedure from job DSNTIJFV.

## 3. FORTRAN LINK EDIT

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.VSF2FORT                       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the FORTRAN linkage editor library.

## 4. PL/I COMPILER

|                    |                                               |
|--------------------|-----------------------------------------------|
| Acceptable values: | blank, or valid data set name: see page 2-101 |
| Default:           | none                                          |
| Update:            | cannot change during update                   |
| DSNZPxxx:          | none                                          |

Specify the data set name of the PL/I compiler load module library. If you specify this option, a STEPLIB is added to each job provided by DB2 that uses this compiler.

If PL/I is not installed on your system, do not run jobs DSNTEJ1P, DSNTEJ2P, DSNTEJ3P, DSNTEJ4P, DSNTEJ5P, DSNTEJ6P, or DSNTEJ6S. You can use SPUFI or QMF to provide the listings of the sample tables and dynamic SQL examples that are provided in jobs DSNTEJ1P and DSNTEJ3P. Remove the DSNHPLI procedure from job DSNTIJMV and the statement that renames the procedure from jobs DSNTIJFV and DSNTIJMV.

## 5. PL/I LINK EDIT BASE

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.PLIBASE                        |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the PL/I linkage editor base library.

## 6. PL/I LINK EDIT COMMON

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.SIBMBASE                       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the PL/I linkage editor common library.

## 7. PL/I DYN RUNTIME BASE

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.PLILINK                        |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the PL/I run-time base library.

## 8. PL/I DYN RUNTIME COMMON

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.SIBMLINK                       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the PL/I run-time common library.

## 9. CPP PROCEDURE LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | CBC.V3R1M1.SCBC3UTL                 |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name containing procedures to set up and invoke the C++ compiler.

## 10. LE/370 RUN TIME LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | CEE.V1R5M0.SCEERUN                  |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

This field is used to tailor the JCL procedure that starts the stored procedures address space. If you enter a value in this field but do not enter a value for IBM COBOL COMPILER, this value is used to tailor the JCL procedure for the stored procedures address space only. Be aware that if you are using OS/390 Release 3, you must use the Language Environment Version 1 Release 7 data set (CEE.V1R7M0.SCEERUN).

## Data Set Names Panel 5: DSNTIPW

The entries on this panel establish data set names for other product libraries. The values entered on this panel are edited into all pertinent sample and installation jobs. If you do not have the product, accept the default. The default cannot be blanked out.

Many data set names for other products appear in the jobs. Most of these data sets can be entered on this panel and on installation panel DSNTIPU. These names are shown in Table 49 on page 2-121 as they appear in the jobs shipped with DB2. Change the names of the data sets if they are different at your site.

```
DSNTIPW INSTALL DB2 - DATA SET NAMES PANEL 5
===>

Enter data set names below:
1 SYSTEM MACLIB ===> SYS1.MACLIB
2 SYSTEM PROCEDURES ===> SYS1.PROCLIB
3 SORT LIBRARY ===> SYS1.SORTLIB
4 IMS RESLIB ===>
5 ISPF ISPLINK MODULE ===> ISP.V3R5M0.ISPLOAD

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 35. Data Set Names Panel 5: DSNTIPW

### 1. SYSTEM MACLIB

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.MACLIB                         |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the assembler macro library.

### 2. SYSTEM PROCEDURES

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.PROCLIB                        |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the system procedures library.

### 3. SORT LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.SORTLIB                        |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name where the DFSORT load module resides. If you do not have DFSORT, an equivalent product is required. If your load library is not in the link list, you can change the DSNUPROC JCL procedure in job DSNTIJMV.

### 4. IMS RESLIB

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | none                                |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the IMS linkage editor library.

If you do not have IMS, you can skip connecting DB2 to IMS and the Phase 4 sample application jobs DSNTJ4C and DSNTJ4P.

### 5. ISPF ISPLINK MODULE

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | ISP.V3R5M0.ISPLOAD                  |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name of the ISPF load module library.

## CICS Release Panel: DSNTIPV

The entry on this panel determines whether you have CICS and which default values appear on the next CICS panel, DSNTIP3.

```
DSNTIPV INSTALL DB2 - CICS RELEASE
===> _

Enter data below:

 1 CICS RELEASE ===> 0 0 = not using CICS
 5 = CICS/ESA 3.3.0
 6 = CICS/ESA Version 4
 7 = CICS Transaction Server for OS/390

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 36. CICS Release Panel: DSNTIPV

### 1. CICS RELEASE

|                    |                             |
|--------------------|-----------------------------|
| Acceptable values: | 0, 5, 6, 7                  |
| Default:           | 0                           |
| Update:            | cannot change during update |
| DSNZPxxx:          | none                        |

Specify the CICS release you are using with this DB2 subsystem. If you select option 0, no editing is done on the CICS jobs, but panel DSNTIP3 still appears. If you do not have CICS, skip the sections about connecting DB2 to CICS and the sample application jobs DSNTJ5A, DSNTJ5C, and DSNTJ5P. If you select option 6 or 7, DSNTIJSU is not required and will not be generated.

#  
#

## Install DB2 CICS Datasets Panel: DSNTIP3

```
DSNTIP3 INSTALL DB2 - CICS DATASETS
====>
DSNT444I Scrolling backward may change fields marked with asterisks
Enter CICS library data below:

* 1 CICS COBOL LIBRARY ====>
* 2 CICS COBOL II LIBRARY====>
* 3 CICS PL/I LIBRARY ====>
* 4 CICS LOAD LIBRARY ====>
* 5 CICS MACRO LIBRARY ====>

CICS Resource Definition Online (RDO) data below (REQUIRED):

 6 CSD NAME ====>
 7 LIST NAME ====> CICS RDO list name
 8 DB2 SAMPLES GROUP ====> Group for CICS-DB2 samples
 9 IS SAMPLES GROUP NEW?====> YES or NO

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 37. CICS Datasets Panel: DSNTIP3

### 1. CICS COBOL LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | Depends on CICS release specified   |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name for the CICS library for use by the COBOL programs.

### 2. CICS COBOL II LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | SYS1.COB2CICS                       |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name for the CICS library shipped with COBOL II or COBOL/370.

### 3. CICS PL/I LIBRARY

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | Depends on CICS release specified   |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name for the CICS library for use by the PL/I programs.



#### 4. CICS LOADLIB

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | Depends on CICS release specified   |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name for the CICS load module library.

#### 5. CICS MACLIB

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | Depends on CICS release specified   |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name for the CICS macro library.

#### 6. CSD NAME

|                    |                                     |
|--------------------|-------------------------------------|
| Acceptable values: | valid data set name: see page 2-101 |
| Default:           | none                                |
| Update:            | cannot change during update         |
| DSNZPxxx:          | none                                |

Specify the data set name for your CICS system definition (CSD) file.

#### 7. LIST NAME

|                    |                             |
|--------------------|-----------------------------|
| Acceptable values: | valid name                  |
| Default:           | none                        |
| Update:            | cannot change during update |
| DSNZPxxx:          | none                        |

Specify the name of the CICS RDO list that the group name from field 10 will be added to. This CICS list name must be the list name used in the CICS startup parameters, or must be added to that list. The name you specify for this field is edited into job DSNTIJSU.

#### 8. DB2 SAMPLES GROUP

|                    |                             |
|--------------------|-----------------------------|
| Acceptable values: | valid name                  |
| Default:           | none                        |
| Update:            | cannot change during update |
| DSNZPxxx:          | none                        |

Enter the CICS RDO group name for the DB2 samples.

#### 9. IS DB2 SAMPLES GROUP NEW?

|                    |                             |
|--------------------|-----------------------------|
| Acceptable values: | YES, NO                     |
| Default:           | none                        |
| Update:            | cannot change during update |
| DSNZPxxx:          | none                        |

Indicate whether the name of the samples group for CICS RDO is new. Specify NO if the group name already exists. If you specify YES, job DSNTIJSU is

automatically edited to add the group. If you specify NO, job DSNTIJSU first deletes the group and then adds it as new.

---

## Sizes Panel: DSNTIPD

The entries on this panel establish the size of the DB2 catalog, directory, work file database, and log data sets.

The values you supply on this panel are estimates used in calculating sizes for main storage and data sets. The values do not reduce any system limits and do not preclude an application or user from exceeding these estimates, within reasonable limits. For example, if you specify 500 databases, you could create 600. However, if you exceed the values by a large margin, you might encounter a shortage of main storage or use many secondary extents for some data sets. The main storage values can usually be changed using the next panel. If the values cannot be changed with the update process, see “The Update Process” on page 2-228 for the appropriate method.

The installation CLIST contains formulas that calculate the space needed for each catalog data set and the indexes required for each data set. Data entered on this panel is used in these formulas. Use integers; do not enter fractions. You can use K (as in 32K) for multiples of 1024 bytes and M (as in 16M) for multiples of 1 048 576 bytes in any field, provided you do not exceed the maximum accepted by the field. For example, for field 10, which has a maximum of 32 000, you can enter 31K, meaning 31 744 bytes. Values of 32K and above exceed the maximum acceptable value for this field.

Many of the fields on this panel affect the value of the EDMPOOL parameter in macro DSN6SPRM.

The defaults for most of the parameters on this panel correspond to the medium storage sizes for the site models shown in Table 8 on page 2-16. One exception is the value for the work file database. To obtain this value, see “DASD Requirements for the Work File Database” on page 2-21. If you have a large site, increase the values according to your needs.

**If you are migrating**, DB2 for OS/390 Version 5 adopts your DB2 for MVS/ESA Version 4 DB2 catalog, directory, work file databases, BSDS, and active logs. Therefore, during migration, you cannot change any of the fields on this panel that affect those data sets.

**Updating the parameters:** You can alter the characteristics of the DB2 catalog, directory, work file databases, BSDS, and active and archive logs by using the methods described on page 2-228. You cannot actually change the values of these parameters.

```

DSNTIPD INSTALL DB2 - SIZES
====> _

Check numbers and reenter to change:

 1 DATABASES ===> 200 In this subsystem
 2 TABLES ===> 10 Per database (average)
 3 COLUMNS ===> 10 Per table (average)
 4 VIEWS ===> 3 Per table (average)
 5 TABLE SPACES ===> 10 Per database (average)
 6 PLANS ===> 200 In this subsystem
 7 PLAN STATEMENTS ===> 30 SQL statements per plan (average)
 8 PACKAGES ===> 300 In this subsystem
 9 PACKAGE STATEMENTS ===> 10 SQL statements per package (average)
10 PACKAGE LISTS ===> 2 Package lists per plan (average)
11 EXECUTED STMTS ===> 15 SQL statements executed (average)
12 TABLES IN STMT ===> 2 Tables per SQL statement (average)
13 TEMP 4K SPACE ===> 16M Bytes of 4K-page work space
14 TEMP 4K DATA SETS ===> 1 Number of data sets for 4K data
15 TEMP 32K SPACE ===> 4M Bytes of 32K-page work space
16 TEMP 32K DATA SETS ===> 1 Number of data sets for 32K data

PRESS: ENTER to continue RETURN to exit HELP for more information

```

Figure 38. Sizes Panel: DSNTIPD

### 1. DATABASES

Acceptable values: 1-64000  
 Default: 200  
 Update: see Update on page 2-228; not during migration  
 DSNZPxxx: none

Estimate the number of user databases in your subsystem.

### 2. TABLES

Acceptable values: 1-400  
 Default: 10  
 Update: see Update on page 2-228; not during migration  
 DSNZPxxx: none

Estimate the average number of tables per database in your subsystem.

### 3. COLUMNS

Acceptable values: 1-750  
 Default: 10  
 Update: see Update on page 2-228; not during migration  
 DSNZPxxx: none

Estimate the average number of columns per table in your subsystem.

#### 4. VIEWS

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | 1-200                    |
| Default:           | 3                        |
| Update:            | see Update on page 2-228 |
| DSNZPxxx:          | none                     |

Estimate the average number of views per table in your subsystem.

#### 5. TABLE SPACES

|                    |                                                |
|--------------------|------------------------------------------------|
| Acceptable values: | 1-400                                          |
| Default:           | 10                                             |
| Update:            | see Update on page 2-228; not during migration |
| DSNZPxxx:          | none                                           |

Estimate the average number of table spaces per database in your subsystem.

#### 6. PLANS

|                    |                                                |
|--------------------|------------------------------------------------|
| Acceptable values: | 1-32000                                        |
| Default:           | 200                                            |
| Update:            | see Update on page 2-228; not during migration |
| DSNZPxxx:          | none                                           |

Estimate the number of application plans in your subsystem. Each program requires a separate application plan.

#### 7. PLAN STATEMENTS

|                    |                                                |
|--------------------|------------------------------------------------|
| Acceptable values: | 1-32000                                        |
| Default:           | 30                                             |
| Update:            | see Update on page 2-228; not during migration |
| DSNZPxxx:          | none                                           |

Estimate the average number of SQL statements per application plan.

#### 8. PACKAGES

|                    |                                                |
|--------------------|------------------------------------------------|
| Acceptable values: | 1-256000                                       |
| Default:           | 300                                            |
| Update:            | see Update on page 2-228; not during migration |
| DSNZPxxx:          | none                                           |

Estimate the total number of packages in the system.

#### 9. PACKAGE STATEMENTS

|                    |                                                |
|--------------------|------------------------------------------------|
| Acceptable values: | 1-32000                                        |
| Default:           | 10                                             |
| Update:            | see Update on page 2-228; not during migration |
| DSNZPxxx:          | none                                           |

Estimate the number of individual SQL statements per package.

## 10. PACKAGE LISTS

|                    |                                                |
|--------------------|------------------------------------------------|
| Acceptable values: | 1-32000                                        |
| Default:           | 2                                              |
| Update:            | see Update on page 2-228; not during migration |
| DSNZPxxx:          | none                                           |

Estimate the average number of packages in a package list per plan.

## 11. EXECUTED STMTS

|                    |                                                |
|--------------------|------------------------------------------------|
| Acceptable values: | 1-32000                                        |
| Default:           | 15                                             |
| Update:            | see Update on page 2-228; not during migration |
| DSNZPxxx:          | none                                           |

Estimate the average number of SQL statements executed per plan. The number of SQL statements executed can be less than the number written.

## 12. TABLES IN STMT

|                    |                                                |
|--------------------|------------------------------------------------|
| Acceptable values: | 1-16                                           |
| Default:           | 2                                              |
| Update:            | see Update on page 2-228; not during migration |
| DSNZPxxx:          | none                                           |

Estimate the average number of tables used per SQL statement. Some SQL statements use more than one table—for example, those using joins, unions, or subselect clauses. Consider how often you expect to use such statements when choosing a value for this parameter.

## 13. TEMP 4K SPACE

|                    |                                 |
|--------------------|---------------------------------|
| Acceptable values: | 1M-2000M                        |
| Default:           | 16M                             |
| Update:            | see below; not during migration |
| DSNZPxxx:          | none                            |

Specify the total size of the 4KB table spaces in the work file database.

Database DSND B07 is used as temporary space for SQL statements that require working storage. In particular, this includes statements that use:

|                                    |                  |
|------------------------------------|------------------|
| GROUP BY or HAVING (without index) | IN (subselect)   |
| ORDER BY (without index)           | ANY (subselect)  |
| DISTINCT (without index)           | SOME (subselect) |
| UNION (except UNION ALL)           | ALL (subselect)  |
| EXISTS (subselect)                 | Some joins       |

Fields 13 and 15 allow you to create two kinds of table spaces within the work file database: one for 4KB pages and one for 32KB pages. The names of the data sets for these table spaces are:

- DSNCAT.DSNDBD.DSNDB07.DSN4Kxx.I0001.A001 for 4KB pages
- DSNCAT.DSNDBD.DSNDB07.DSN32Kxx.I0001.A001 for 32KB pages

where *xx* is the number of the table space.

The space specified on this parameter is divided equally among each of the temporary 4KB table spaces. For example, if you specify 16M for the TEMP 4K SPACE field and 4 for the TEMP 4K DATA SETS field, then each 4KB temporary data set is allocated 4MB of space.

Because device characteristics differ, the actual amount of space may vary.

All DB2 users share those table spaces. Utilities cannot be used on them.

You can create additional temporary table spaces at any time except during migration. This improves DB2 performance by reducing device contention among applications that require working storage. You can also concatenate temporary table spaces to support large temporary files. For information about creating additional temporary table spaces, see Section 5 (Volume 2) of *Administration Guide*.

**Updating:** You can change the size of the data sets by deleting and redefining them when DB2 is not running or when the work file database is stopped. Job DSNTIJTM is a useful example. For information on job DSNTIJTM, see page 2-252.

#### 14. TEMP 4K DATA SETS

|                    |                                    |
|--------------------|------------------------------------|
| Acceptable values: | 1-99                               |
| Default:           | 1                                  |
| Update:            | see field 13; not during migration |
| DSNZPxxx:          | none                               |

Estimate the number of temporary data sets for 4KB pages.

#### 15. TEMP 32K SPACE

|                    |                                                 |
|--------------------|-------------------------------------------------|
| Acceptable values: | 0-2000M                                         |
| Default:           | 4M                                              |
| Update:            | see field 11 or see below; not during migration |
| DSNZPxxx:          | none                                            |

Specify the total size of the 32KB table spaces in the work file database.

If you do not use 32KB buffers, enter 0. Then, enter 0 for the BP32K*n* fields on the Buffer Pool Sizes Panel 2 (DSNTIP2).

Because device characteristics differ, the actual amount of space

**Updating:** To update this field, create a 32KB table space for the work file database. To create a 32KB table space, follow these steps:

1. Stop DB2.
2. Run the CLIST in install mode. Be sure you enter positive values for fields TEMP 32K SPACE, TEMP 32K DATA SETS, and BP32K on the Buffer Pool Sizes Panel 2 (DSNTIP2).

3. Run job DSNTIJUZ to update the subsystem parameter values.
4. Start DB2. Make sure you do not access the 32KB table space until you complete the next step. To control access, use the following command:  
-DSN1 START DB2 ACCESS(MAINT)
5. Edit the DSNTIJTM job as follows:
  - Delete everything except the DSNTIC procedure and steps DSNTTMP and DSNTIST.
  - Delete the control statements that define the 4KB data set in DSNTTMP.
  - Delete the control statements that define the 4KB table space in step DSNTIST.
  - Remove the step names that do not apply from the COND statements in the JCL.
  - Execute the job.

#### 16. TEMP 32K DATA SETS

|                    |                                           |
|--------------------|-------------------------------------------|
| Acceptable values: | 0-99                                      |
| Default:           | 1                                         |
| Update:            | see fields 13 or 15; not during migration |
| DSNZPxxx:          | none                                      |

Specify the number of temporary data sets for 32KB pages.



## Thread Management Panel: DSNTIPE

The entries on this panel determine main storage sizes.

**Updating the parameters:** You can use UPDATE mode of the CLIST to update any value on this panel.

```
DSNTIPE INSTALL DB2 - THREAD MANAGEMENT
====> _

Check numbers and reenter to change:

 1 DATABASES ====> 100 Concurrently in use
 2 MAX USERS ====> 70 Concurrently running in DB2
 3 MAX REMOTE ACTIVE ====> 64 Maximum number of active
 database access threads
 4 MAX REMOTE CONNECTED ====> 64 Maximum number of active/inactive
 database access threads
 5 MAX TSO CONNECT ====> 40 Users on QMF or in DSN command
 6 MAX BATCH CONNECT ====> 20 Users in DSN command or utilities
 7 SEQUENTIAL CACHE ====> BYPASS 3990 Storage for sequential IO
 Values are SEQ or BYPASS
 8 UTILITY CACHE OPTION ====> NO 3990 storage for DB2 utility IO
 9 DEFAULT INDEX TYPE ====> 2 1 or 2
10 MAX KEPT DYN STMTS ====> 5000 Maximum number of prepared dynamic
 statements saved past commit points

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 39. Thread Management Panel: DSNTIPE

### 1. DATABASES

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1-800                      |
| Default:           | 100                        |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | none                       |

Specify the maximum number of databases that can be open at one time. The number is affected primarily by DSMAX on panel DSNTIPC, which specifies the number of open datasets. See “Install DB2—CLIST Calculations Panel 1: DSNTIPC” on page 2-215 for more information about DSMAX. The number is also affected by the CLOSE clause of CREATE TABLESPACE and CREATE INDEX statements, as well as by START and STOP commands. For instance, you might want to specify a smaller value if you use CLOSE YES extensively. For performance considerations, see Section 5 (Volume 2) of *Administration Guide*.

### 2. MAX USERS

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1-2000                     |
| Default:           | 70                         |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP CTHREAD           |

Specify the maximum number of allied threads (threads started at the local subsystem) that can be allocated concurrently. Count the following as separate users:

#

- Each TSO user (whether running a DSN command or a DB2 request from QMF)
- Each batch job (whether running a DSN command or a DB2 utility)
- Each IMS region that can access DB2
- Each active CICS transaction that can access DB2
- Each task connected to DB2 through the call attachment facility.

The total number of threads accessing data that can be allocated concurrently is the sum of the MAX USERS value and field 3, MAX REMOTE ACTIVE. The maximum allowable value for this sum is 2000. When the number of users attempting to access DB2 exceeds the number you specify, excess plan allocation requests are queued.

### 3. MAX REMOTE ACTIVE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 0-1999                     |
| Default:           | 64                         |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP MAXDBAT           |

Specify the maximum number of active database access threads (DBATs) that can be allocated concurrently. DBATs are allocated on demand at your local subsystem on behalf of an SQL request started at a remote subsystem.

A value of zero in this field can be used to restrict DDF server activity on a member of a data sharing group. When this field is zero, expect the following:

#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#

- DDF does not register the member's LU name with the VTAM generic LU name during DDF startup. This causes VTAM generic resource connections to be directed to DB2 members that specify a MAX REMOTE ACTIVE value of greater than zero.
- DDF does not register the member with WLM for member-specific sysplex routing. This does not prevent the member from using WLM for enclave prioritization, but it prevents WLM from including this member in the sysplex routing data sent to remote sites.
- DDF does not listen on the DRDA SQL port. This means TCP/IP SQL connections can only be accepted by members that specify MAX REMOTE ACTIVE greater than zero.
- DDF rejects requests for the Sysplex Routing TPN with a *SNA TPN not available* sense code.
- In a non-data sharing environment, when MAX REMOTE ACTIVE is zero, DB2 rejects server connection requests unless the connection request is for resynchronizing existing indoubt units of recovery.

The total number of threads accessing data concurrently is the sum of field 2, MAX USERS, and this field, MAX REMOTE ACTIVE. The maximum allowable value for this sum is 2000. When the limit is reached, all requests to create DBATs are queued.

#### 4. MAX REMOTE CONNECTED

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 0-25000                    |
| Default:           | 64                         |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP CONDBAT           |

Specify the maximum number of database access threads (DBATs), both active and inactive, that can be connected to this DB2 subsystem concurrently. See Section 5 (Volume 2) of *Administration Guide* for more information about inactive threads.

#### 5. MAX TSO CONNECT

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1-2000                     |
| Default:           | 40                         |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP IDFORE            |

Specify the maximum number of users allowed to be identified to DB2 from TSO foreground at the same time. Count each of the following as a separate user:

- Each TSO foreground user executing a DSN command.
- Each TSO foreground user connected to DB2 through the call attachment facility (CAF). This can include QMF users running in TSO foreground or user-written CAF applications running in TSO foreground.

When the number of TSO users attempting to access DB2 exceeds the number you specify, excess connection requests are rejected. There is no subsystem parameter that controls the maximum concurrent connections for IMS and CICS. You can control those limits by using IMS and CICS facilities. For the CICS attachment, the maximum number of connections to DB2 can be controlled using the resource control table (RCT) TYPE=INIT THRDMAX value.

#### 6. MAX BATCH CONNECT

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1-2000                     |
| Default:           | 40                         |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP IDBACK            |

Specify the maximum number of concurrent connections identified to DB2 from batch. Count each of the following as a separate connection:

- Each batch job using QMF
- Each batch job using the DSN command processor
- Each task connected to DB2 through the call attach facility running in batch. Among others, this can include:
  - Batch jobs using QMF
  - APPC applications
  - TCP/IP FTP connections.

Requests to access DB2 by batch jobs that exceed this limit are rejected.

## 7. SEQUENTIAL CACHE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | BYPASS, SEQ                |
| Default:           | BYPASS                     |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM SEQCACH           |

Specify whether to use the sequential mode to read cached data from a 3990 controller. If you accept the default, BYPASS, DB2 prefetch will bypass the cache. If you specify SEQ, DB2 prefetch will use sequential access for read activity. Many sites gain a performance benefit by specifying SEQ and using DFSMS or DFP controls with newer 3990 caches. See Section 5 (Volume 2) of *Administration Guide* for a discussion of these considerations.

## 8. UTILITY CACHE OPTION

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | YES, NO                    |
| Default:           | NO                         |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM SEQPRES           |

Specify whether utilities that do a scan of a nonpartitioned index followed by an update of a subset of the pages in the index allow data to remain in 3990 cache longer when reading data. If you specify YES, these DB2 utility prefetch reads remain in cache longer, possibly improving performance of subsequent writes in the following cases for a table with very large nonpartitioned indexes:

LOAD PART integer RESUME  
REORG TABLESPACE PART

This option is useful only with RAMAC DASD attached to the 3990 Model 6.

If you specify NO, DB2 utilities use the 3990 cache the same way as any other application (as you specified in the SEQUENTIAL CACHE option).

## 9. DEFAULT INDEX TYPE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1, 2                       |
| Default:           | 2                          |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM DEFIXTP           |

Specify the default index type to be used when the TYPE option is omitted from a CREATE INDEX statement. DEFIXTP is set to 1 or 2, depending on the index type.

## 10. MAX KEPT DYN STMTS

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 0-65535                    |
| Default:           | 5000                       |
| Update:            | option 12 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM MAXKEEPD          |

Specify the total number of prepared, dynamic SQL statements to save past a commit point by applications running with the KEEP DYNAMIC(YES) bind option. This is a system-wide limit. This parameter does not limit the size of the dynamic cache itself.

# When many applications bound with KEEP\_DYNAMIC(YES) are run in a system that  
# has the dynamic statement cache active, they can use a considerable amount of  
# storage in the DBM1 address space. This parameter helps limit the amount of  
# storage used by these applications by limiting the total number of prepared  
# statements held by these applications past a commit point. If this limit is exceeded,  
# the KEEP\_DYNAMIC(YES) behavior will be honored by DB2, but "implicit" prepares  
# may be necessary to rebuild the executable version of some SQL statements when  
# they are executed after a commit. For more information about the interaction of the  
# KEEP\_DYNAMIC(YES) bind option and the dynamic statement cache, refer to  
# Section 6 of *Application Programming and SQL Guide*.

# When you enter 0, DB2 can not keep the executable version of dynamic SQL  
# statements past commit points. To retain the KEEP\_DYNAMIC(YES) behavior after a  
# commit point, DB2 performs "implicit" prepares to rebuild the executable version of  
# the dynamic SQL statements.

## Buffer pool Sizes Panel 1: DSNTIP1

This is the first of two panels that lets you choose the size of your virtual buffer pools, the size of your hiperpools, and whether or not MVS can steal the expanded storage used by DB2 data stored in a hiperpool. For information about the second buffer pool sizes panel, see “Buffer pool Sizes Panel 2: DSNTIP2” on page 2-152. For a complete description of tuning buffer pools and hiperpools, see Section 5 (Volume 2) of *Administration Guide*.

**Updating the buffer pool sizes:** You can change your buffer pool sizes online with the ALTER BUFFERPOOL command, but you cannot change these sizes by running the DSNTINST CLIST in update mode.

```

DSNTIP1 INSTALL DB2 - BUFFER POOL SIZES PANEL 1
====> _
Enter sizes (in number of pages). Specify yes (Y) or no (N) for CASTOUT.

 BUFFERPOOL Hiperpool CASTOUT
1 BP0 ==> 2000 ==> 0 ==> Y 16 BP15 ==> 0 ==> 0 ==> Y
2 BP1 ==> 0 ==> 0 ==> Y 17 BP16 ==> 0 ==> 0 ==> Y
3 BP2 ==> 0 ==> 0 ==> Y 18 BP17 ==> 0 ==> 0 ==> Y
4 BP3 ==> 0 ==> 0 ==> Y 19 BP18 ==> 0 ==> 0 ==> Y
5 BP4 ==> 0 ==> 0 ==> Y 20 BP19 ==> 0 ==> 0 ==> Y
6 BP5 ==> 0 ==> 0 ==> Y 21 BP20 ==> 0 ==> 0 ==> Y
7 BP6 ==> 0 ==> 0 ==> Y 22 BP21 ==> 0 ==> 0 ==> Y
8 BP7 ==> 0 ==> 0 ==> Y 23 BP22 ==> 0 ==> 0 ==> Y
9 BP8 ==> 0 ==> 0 ==> Y 24 BP23 ==> 0 ==> 0 ==> Y
10 BP9 ==> 0 ==> 0 ==> Y 25 BP24 ==> 0 ==> 0 ==> Y
11 BP10 ==> 0 ==> 0 ==> Y 26 BP25 ==> 0 ==> 0 ==> Y
12 BP11 ==> 0 ==> 0 ==> Y 27 BP26 ==> 0 ==> 0 ==> Y
13 BP12 ==> 0 ==> 0 ==> Y 28 BP27 ==> 0 ==> 0 ==> Y
14 BP13 ==> 0 ==> 0 ==> Y 29 BP28 ==> 0 ==> 0 ==> Y
15 BP14 ==> 0 ==> 0 ==> Y 30 BP29 ==> 0 ==> 0 ==> Y

PRESS: ENTER to continue RETURN to exit HELP for more information

```

Figure 40. Buffer Pool Sizes Panel 1: DSNTIP1

### 1-30. BUFFERPOOL

|                    |                                            |
|--------------------|--------------------------------------------|
| Acceptable values: | for BP0, 56-400000; for BP1-BP29, 0-400000 |
| Default:           | for BP0, 2000; for BP1-BP29, 0             |
| Update:            | ALTER BUFFERPOOL command                   |
| DSNZPxxx:          | none                                       |

Specify the total number of 4KB buffers in the given virtual buffer pool (BP0-BP29).

### 1-30. Hiperpool

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | 0-2097152                |
| Default:           | 0                        |
| Update:            | ALTER BUFFERPOOL command |
| DSNZPxxx:          | none                     |

Specify the size (in number of pages) of the hiperpool associated with the given virtual buffer pool (BP0-BP29). If you specify a size in this field, a hiperpool will be

created as an extension to that virtual buffer pool. If the size of a given hiperpool is greater than zero, the size of its corresponding buffer pool must also be greater than zero. The total size of all hiperpools must not exceed 8GB.

### 1-30. **CASTOUT**

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | Y, N                     |
| Default:           | Y                        |
| Update:            | ALTER BUFFERPOOL command |
| DSNZPxxx:          | none                     |

Specify whether MVS can steal the expanded storage used by DB2 data in a hiperpool associated with one of the virtual buffer pools, BP0-BP29.

'Y' indicates that MVS can discard the copy of the DB2 data stored in the hiperpool when MVS needs the expanded storage or when the hiperpool usage is low.

'N' indicates that MVS should give the copy of the DB2 data stored in the hiperpool a higher priority before discarding the data. MVS could still steal the expanded storage if it needs the space, but can discard the data if one of the following conditions are met:

- The hiperspace is deleted.
- The hiperspace storage area is explicitly released.
- The expanded storage pages need to be reconfigured because of MVS hiperspace maintenance.

## Buffer pool Sizes Panel 2: DSNTIP2

This is the second of the two panels that lets you choose the size of your virtual buffer pools, the size of your hiperpools, and whether or not MVS can steal the expanded storage used by DB2 data stored in a hiperpool. The first panel is described in "Buffer pool Sizes Panel 1: DSNTIP1" on page 2-150. For a complete description of tuning buffer pools and hiperpools, see Section 5 (Volume 2) of *Administration Guide*.

**Updating the buffer pool sizes:** You can change your buffer pool sizes online with the ALTER BUFFERPOOL command, but you cannot change these sizes by running the DSNTINST CLIST in update mode.

```

DSNTIP2 INSTALL DB2 - BUFFER POOL SIZES PANEL 2
====> _
Enter sizes (in number of pages). Specify yes (Y) or no (N) for CASTOUT.

 BUFFERPOOL Hiperpool CASTOUT BUFFERPOOL Hiperpool CASTOUT
1 BP30 ==> 0 ==> 0 ==> Y 16 BP45 ==> 0 ==> 0 ==> Y
2 BP31 ==> 0 ==> 0 ==> Y 17 BP46 ==> 0 ==> 0 ==> Y
3 BP32 ==> 0 ==> 0 ==> Y 18 BP47 ==> 0 ==> 0 ==> Y
4 BP33 ==> 0 ==> 0 ==> Y 19 BP48 ==> 0 ==> 0 ==> Y
5 BP34 ==> 0 ==> 0 ==> Y 20 BP49 ==> 0 ==> 0 ==> Y
6 BP35 ==> 0 ==> 0 ==> Y 21 BP32K ==> 24 ==> 0 ==> Y
7 BP36 ==> 0 ==> 0 ==> Y 22 BP32K1 ==> 0 ==> 0 ==> Y
8 BP37 ==> 0 ==> 0 ==> Y 23 BP32K2 ==> 0 ==> 0 ==> Y
9 BP38 ==> 0 ==> 0 ==> Y 24 BP32K3 ==> 0 ==> 0 ==> Y
10 BP39 ==> 0 ==> 0 ==> Y 25 BP32K4 ==> 0 ==> 0 ==> Y
11 BP40 ==> 0 ==> 0 ==> Y 26 BP32K5 ==> 0 ==> 0 ==> Y
12 BP41 ==> 0 ==> 0 ==> Y 27 BP32K6 ==> 0 ==> 0 ==> Y
13 BP42 ==> 0 ==> 0 ==> Y 28 BP32K7 ==> 0 ==> 0 ==> Y
14 BP43 ==> 0 ==> 0 ==> Y 29 BP32K8 ==> 0 ==> 0 ==> Y
15 BP44 ==> 0 ==> 0 ==> Y 30 BP32K9 ==> 0 ==> 0 ==> Y

PRESS: ENTER to continue RETURN to exit HELP for more information

```

Figure 41. Buffer Pool Sizes Panel 2: DSNTIP2

### 1-30. BUFFERPOOL

|                    |                                                          |
|--------------------|----------------------------------------------------------|
| Acceptable values: | for BP30-BP49, 0-400000; for<br>BP32K-BP32K9, 0-50000    |
| Default:           | for BP30-BP49, 0; for BP32K, 24; for<br>BP32K1-BP32K9, 0 |
| Update:            | ALTER BUFFERPOOL command                                 |
| DSNZPxxx:          | none                                                     |

Specify the total number or 4KB buffers in the given virtual buffer pool (BP30-BP49), or specify the number of 32K buffers in the given virtual buffer pool (BP32K-BP32K9).

### 1-30. Hiperpool



|                    |                                                            |
|--------------------|------------------------------------------------------------|
| Acceptable values: | 0-2097152 for 4KB hiperpools, 0-262144 for 32KB hiperpools |
| Default:           | 0                                                          |
| Update:            | ALTER BUFFERPOOL command                                   |
| DSNZPxxx:          | none                                                       |

Specify the size (in number of pages) of the hiperpool associated with the given virtual buffer pool (BP30-BP49 or BP32K-BP32K9). If you specify a size in this field, a hiperpool will be created as an extension to that virtual buffer pool. If the size of a given hiperpool is greater than zero, the size of its corresponding buffer pool must also be greater than zero. The total size of all hiperpools must not exceed 8GB.

### 1-30. **CASTOUT**

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | Y, N                     |
| Default:           | Y                        |
| Update:            | ALTER BUFFERPOOL command |
| DSNZPxxx:          | none                     |

Specify whether MVS can steal the expanded storage used by DB2 data in a hiperpool associated with one of the virtual buffer pools, BP30-BP49 or BP32K-BP32K9.

'Y' indicates that MVS can discard the copy of the DB2 data stored in the hiperpool when MVS needs the expanded storage or when the hiperpool usage is low.

'N' indicates that MVS cannot discard the copy of the DB2 data stored in the hiperpool unless one of the following conditions are met:

- The hiperspace is deleted
- The hiperspace storage area is explicitly released
- The expanded storage pages need to be reconfigured because of MVS hiperspace maintenance.

## Tracing Panel: DSNTIPN

The entries on this panel affect the audit, global, accounting, and monitor traces as well as checkpoint frequency. For more information on these trace categories, see Section 5 (Volume 2) of *Administration Guide*.

```
DSNTIPN INSTALL DB2 - TRACING
===> _

Enter data below:

 1 AUDIT TRACE ===> NO Audit classes to start. NO,YES,list
 2 TRACE AUTO START ===> NO Global classes to start. YES,NO,list
 3 TRACE SIZE ===> 64K Trace table size in bytes. 4K-396K
 4 SMF ACCOUNTING ===> 1 Accounting classes to start. NO,YES,list
 5 SMF STATISTICS ===> YES Statistics classes to start. NO,YES,list
 6 STATISTICS TIME ===> 30 Time interval in minutes. 1-1440

 7 MONITOR TRACE ===> NO Monitor classes to start. NO,YES,list
 8 MONITOR SIZE ===> 8K Default monitor buffer size. 8K-1M

 9 CHECKPOINT FREQ ===> 50000 Number of log records per checkpoint
10 UR CHECK FREQ ===> 0 Checkpoints required to enable UR check.
 0-255

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 42. Tracing Panel: DSNTIPN

### 1. AUDIT TRACE

|                    |                                           |
|--------------------|-------------------------------------------|
| Acceptable values: | YES, NO, list of classes, an asterisk (*) |
| Default:           | NO                                        |
| Update:            | option 15 on panel DSNTIPB                |
| DSNZPxxx:          | DSN6SYSP AUDITST                          |

Specify whether to start the audit trace automatically when DB2 is started, and specify the classes for which to start it.

NO specifies no automatic start; if the audit trace is to be used, it must be started with the START TRACE command. YES starts the trace for the default class (class 1) whenever DB2 is started. To specify other classes for which trace must start automatically, list the numbers (any integer from 1 to 32) separated by commas. Only classes 1-9 are defined by DB2. Enter an asterisk (\*) to start audit trace for all classes. For information on audit classes and the effect of the audit trace, see Section 3 (Volume 1) of *Administration Guide*.

### 2. TRACE AUTO START

|                    |                                           |
|--------------------|-------------------------------------------|
| Acceptable values: | YES, NO, list of classes, an asterisk (*) |
| Default:           | NO                                        |
| Update:            | option 15 on panel DSNTIPB                |
| DSNZPxxx:          | DSN6SYSP TRACSTR                          |

Specify whether to start the global trace automatically when DB2 is started, and specify the classes for which to start it.

NO specifies no automatic start; if the trace is to be used, it must be started with the START TRACE command. YES starts the global trace for the default classes (classes 1, 2, and 3) whenever DB2 is started, and performs additional data consistency checks whenever a data page or index page is modified. To start specific classes, enter a list of class numbers (any integer from 1 to 32) separated by commas. Only classes 1-9 are defined by DB2. Enter an asterisk (\*) to start global trace for all classes.

The global trace is used to diagnose problems in DB2. Customers with production systems requiring high performance might consider turning off global trace. However, be aware that turning off global trace presents a serviceability exposure. In the event of a system failure, IBM service personnel could request that you turn on global trace and attempt to re-create the problem.

### 3. TRACE SIZE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 4K-396K                    |
| Default:           | 64K                        |
| Update:            | option 15 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP TRACTBL           |

Specify the size, in bytes, of the RES trace table. This table is the default destination for the global trace records in DB2. Most trace records require 32-byte entries; events with more than three data items require 64-byte entries.

You can use the abbreviation K for multiples of 1024 bytes. The actual value is rounded up to a multiple of 4096 bytes. If you use 50K, for example, the actual table size is 52KB.

In the subsystem parameter, use a multiple of 4. For example, to get a 64KB table, code TRACTBL=16.

### 4. SMF ACCOUNTING

|                    |                                           |
|--------------------|-------------------------------------------|
| Acceptable values: | YES, NO, list of classes, an asterisk (*) |
| Default:           | 1                                         |
| Update:            | option 15 on panel DSNTIPB                |
| DSNZPxxx:          | DSN6SYSP SMFACCT                          |

Specify whether DB2 sends accounting data to SMF automatically when DB2 is started. This field also specifies what classes are sent.

NO specifies no automatic start. YES starts the trace for the default class (class 1). If you use YES, you can also need to update the SMFPRMxx member of SYS1.PARMLIB to permit SMF to write the records.

To start specific classes, enter a list of class numbers (any integer from 1 to 32) separated by commas. Only classes 1-5 and classes 7,8 are defined by DB2. To start all classes, enter an asterisk (\*). For information on accounting classes, see Section 5 (Volume 2) of *Administration Guide*. For information on using the trace, see "Installation Step 7: Record DB2 Data to SMF (Optional)" on page 2-242.

## 5. SMF STATISTICS

|                    |                                           |
|--------------------|-------------------------------------------|
| Acceptable values: | YES, NO, list of classes, an asterisk (*) |
| Default:           | YES                                       |
| Update:            | option 15 on panel DSNTIPB                |
| DSNZPxxx:          | DSN6SYSP SMFSTAT                          |

Specify whether DB2 sends statistical data to SMF automatically when DB2 is started. This field also specifies what classes are sent.

NO specifies no automatic start. YES starts the trace for the default classes (classes 1, 3, 4, and 5). If you specify YES, you might also need to update the SMFPRMxx member of SYS1.PARMLIB to permit SMF to write the records.

To start specific classes, enter a list of class numbers (any integer from 1 to 32) separated by commas. Only classes 1-5 are defined by DB2. To start all classes, enter an asterisk (\*). For information on statistics classes, see Section 5 (Volume 2) of *Administration Guide*. For information on using the trace, see "Installation Step 7: Record DB2 Data to SMF (Optional)" on page 2-242.

## 6. STATISTICS TIME

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1-1440                     |
| Default:           | 30                         |
| Update:            | option 15 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP STATIME           |

Specify the time interval, in minutes, between statistics collections. Statistics records are written approximately at the end of this interval.

## 7. MONITOR TRACE

|                    |                                           |
|--------------------|-------------------------------------------|
| Acceptable values: | YES, NO, list of classes, an asterisk (*) |
| Default:           | NO                                        |
| Update:            | option 15 on panel DSNTIPB                |
| DSNZPxxx:          | DSN6SYSP MON                              |

Specify whether to start the monitor trace automatically when DB2 is started. This field also specifies what classes are sent.

NO specifies no automatic start. YES starts the trace for the default classes (class 1) whenever DB2 is started. To start the trace automatically for other classes, enter a list of class numbers (any integer from 1 to 32) separated by commas. Only classes 1-8 are defined by DB2. To start all classes, enter an asterisk (\*). For information on the monitor trace, see Appendix G (Volume 2) of *Administration Guide*.

## 8. MONITOR SIZE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 8K-1M                      |
| Default:           | 8K                         |
| Update:            | option 15 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP MONSIZE           |

Specify the default buffer size for monitor trace when sending data to monitor destinations. You can enter the value in bytes (for example, 8192) or use the abbreviation K for kilobytes (for example, 8K).

## 9. CHECKPOINT FREQ

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 200-16000000               |
| Default:           | 50000                      |
| Update:            | option 15 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP LOGLOAD           |

Specify the number of log records that DB2 writes between the start of successive checkpoints. If you accept the default (50 000), DB2 starts a new checkpoint every time 50 000 log records have been written. The number of log records produced depends on several factors, including application mix and complexity, processor and DASD configurations and speeds, and acceptable trade-offs between performance and restart time.

## 10. UR CHECK FREQ

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 0-255                      |
| Default:           | 0                          |
| Update:            | option 15 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP URCHKTH           |

Specify the number of checkpoint cycles to complete before DB2 issues a warning message to the console and instrumentation for an uncommitted unit of recovery (UR).

Accept the default to disable this option. This option does not affect performance. If you use this option, specify a value that is based on how often a checkpoint occurs in your system and how much time you can allow for a restart or shutdown. For example, if your site's checkpoint interval is 5 minutes and the standard limit for issuing COMMITs with units of recovery is 20 minutes, divide 20 by 5 to determine the best value for your system.

## Operator Functions Panel: DSNTIPO

The entries on this panel affect various operator functions, such as write-to-operator route codes, automatic recall, and the maximum amount of CPU time allocated for a dynamic SQL statement.

```
DSNTIPO INSTALL DB2 - OPERATOR FUNCTIONS
===> _

Enter data below:

 1 WTO ROUTE CODES ===> 1 Routing codes for WTORS
 2 RECALL DATABASE ===> YES Use DFHSM automatic recall. YES or NO
 3 RECALL DELAY ===> 120 Seconds to wait for automatic recall

 4 RLF AUTO START ===> NO Resource Limit Facility. NO or YES
 5 RLST NAME SUFFIX ===> 01 Resource Limit Spec. Table (RLST)
 6 RLST ACCESS ERROR ===> NOLIMIT Action on RLST access error. Values are:
 NOLIMIT - Run without limit
 NORUN - Do not run at all
 1-5000000 - Limit to CPU service units

 7 PARAMETER MODULE ===> DSNZPARM Name of DB2 subsystem parameter module
 8 AUTO BIND ===> YES Use automatic bind. YES or NO
 9 EXPLAIN PROCESSING ===> YES Explain allowed on autobind? YES or NO
10 DPROP SUPPORT ===> 1 1=NO 2=ONLY 3=ANY
11 SITE TYPE ===> LOCALSITE LOCALSITE or RECOVERYSITE

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 43. Operator Functions Panel: DSNTIPO

### 1. WTO ROUTE CODES

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | see below                  |
| Default:           | 1                          |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP ROUTCDE           |

Specify the MVS console routing codes assigned to messages that are not solicited from a specific console. You can specify from 1 to 16 route codes. You must use at least one code. Separate codes in a list by commas only, not by blanks; for example: 1,3,5,7,9,10,11. For more information on routing codes, refer to *OS/390 MVS Routing and Descriptor Codes*.

### 2. RECALL DATABASE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | YES, NO                    |
| Default:           | YES                        |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM RECALL            |

Specify whether DFSMSHsm automatic recall is performed for DB2 databases. NO indicates that a DB2 table space that has been migrated is considered to be an unavailable resource. It must be recalled explicitly before it can be used by DB2. YES indicates that DFSMSHsm is invoked to recall it automatically.

### 3. RECALL DELAY

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 0-32767                    |
| Default:           | 120                        |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM RECALLD           |

Specify the maximum length of time in seconds that a program can delay for a DFSMSHsm recall. If the recall is not completed within the specified number of seconds, the program receives an error message indicating that the page set is unavailable, but that recall was initiated. If you use 0 and RECALL DATABASE (field 2) is YES, then the recall is performed asynchronously. This field is ignored if the RECALL DATABASE field is NO.

The RECALL DELAY option is not used when running a DB2 utility against a DB2 migrated data set.

### 4. RLF AUTO START

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | YES, NO                    |
| Default:           | NO                         |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP RLF               |

Specify whether the resource limit facility (governor) is automatically started each time DB2 is started. For information about using the governor, see Section 5 (Volume 2) of *Administration Guide*.

### 5. RLST NAME SUFFIX

|                    |                                                                    |
|--------------------|--------------------------------------------------------------------|
| Acceptable values: | any 2 alphanumeric characters; national characters are not allowed |
| Default:           | 01                                                                 |
| Update:            | option 16 on panel DSNTIPB                                         |
| DSNZPxxx:          | DSN6SYSP RLFTBL                                                    |

Specify the suffix of the default resource limit specification table (RLST). The default RLST is used when the resource limit facility (governor) is automatically started or when you start the governor without specifying a suffix.

### 6. RLST ACCESS ERROR

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | NOLIMIT, NORUN, 1-5000000  |
| Default:           | NOLIMIT                    |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP RLFERR            |

Specify what action DB2 takes if the governor encounters a condition that prevents it from accessing the resource limit specification table or if it cannot find a row in the table that applies to the authorization ID, the plan or package name, and the logical unit of work name of the query user.

- NOLIMIT allows all dynamic SQL statements to run without limit.
- NORUN terminates all dynamic SQL statements immediately with a SQL error code.

- A number from 1 to 5 000 000 is the default limit; if the limit is exceeded, the SQL statement is terminated. For guidelines in choosing the default limit, see Section 5 (Volume 2) of *Administration Guide*.

## 7. PARAMETER MODULE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1-8 characters             |
| Default:           | DSNZPARAM                  |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | none                       |

Specify the member name of the load module for DB2 subsystem parameters. The module will reside in library *prefix*.SDSNEXIT. To avoid conflict with members of *prefix*.SDSNLOAD, use DSNZxxx, where xxx is any set of 3 alphanumeric characters. DB2 puts this name in the startup JCL procedure in SYS1.PROCLIB, but you can override this value using the START DB2 command.

## 8. AUTO BIND

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | YES, NO                    |
| Default:           | YES                        |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM ABIND             |

Specify whether plans or packages are automatically rebound. Specifying YES allows automatic rebind operations to be performed at execution time when a plan or package:

- Is invalid (See Section 4 of *Application Programming and SQL Guide* to see why a plan or package is invalid.)
- Was bound in a higher release and is used for the first time in a fallback release
- Was used in a fallback release but is later remigrated and used the first time in the higher release.

Automatic rebinds can improve performance because you do not have to explicitly rebind all plans and packages. They can also hinder performance because they require access to the DB2 directory and catalog.

Specifying NO for this field means that you must explicitly rebind any invalid plan or package before it can be operational.

# If you have APAR PQ03621 applied to your subsystem, macro DSN6SPRM ABIND  
 # has been modified to accept an additional value of COEXIST. To use this value,  
 # you must manually specify COEXIST for macro DSN6SPRM ABIND in job  
 # DSNTIJUZ. Specifying COEXIST allows automatic rebind operations to be  
 # performed in a DB2 data sharing coexistence environment only when the plan or  
 # package:

- is marked "invalid" (the SYSPLAN or SYSPACKAGE column VALID contains 'N', See Section 4 of *Application Programming and SQL Guide* to see why a plan or package becomes invalid), or
- was last bound on DB2 Version 5 and is now running on DB2 Version 4.

# For this case, DB2 will perform a "Version 5 to Version 4" autobind on Version  
 # 4 for the plan or package before running it there. This autobind is performed  
 # only for the first attempt to run the plan or package on DB2 Version 4.



An automatic rebind operation will not be performed on DB2 Version 5 in a data sharing coexistence environment for any Version 5 plan or package for which DB2 last performed a "Version 5 to Version 4" autobind on Version 4 and that plan or package is now run again on Version 5. The plan or package will run on Version 5 as a Version 4 bound plan or package, but no Version 5 only features such as optimization enhancements, improved access paths, or index usage enhancements will be used because the plan or package was not autobound on DB2 Version 5.

# After all members of a data sharing group have been migrated to the same release  
# level, DB2 interprets a value of COEXIST as YES, which allows all types of  
# autobinds to occur.

# The value COEXIST is relevant only for DB2 subsystems in data sharing mode. If  
# COEXIST is specified in a non-data-sharing environment, DB2 ignores the value  
# and uses the default value of YES when determining whether an autobind can be  
# done. YES allows a "Version 4 to Version 5" autobind on Version 5 after a previous  
# "Version 5 to Version 4" autobind completed on Version 4.

# If you have a DB2 data sharing group where some of the DB2 members are at  
# different DB2 release levels, there could be an increase in the rate of automatic  
# rebinds and, subsequently, a degradation in run-time performance for a plan or  
# package if

- # • the plan or package was last bound on DB2 Version 5 and is now run on DB2  
# Version 4 (a "Version 5 to Version 4" autobind occurs on Version 4), or
- # • DB2 last performed a "Version 5 to Version 4" autobind for that plan or  
# package on DB2 Version 4, and the plan or package is now run on Version 5  
# (a "Version 4 to Version 5" autobind occurs on Version 5).

# To reduce the rate of automatic rebinds in this type of data sharing consider  
# specifying COEXIST for AUTO BIND.

## 9. EXPLAIN PROCESSING

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | YES, NO                    |
| Default:           | YES                        |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM ABEXP             |

Specify whether you want EXPLAIN processing to occur during automatic rebind.

'YES' specifies that you want EXPLAIN processing to occur during automatic rebind of a plan or package when the bind option EXPLAIN(YES) is specified. If the PLAN\_TABLE does not exist, automatic rebind continues, but there is no EXPLAIN output. If you specified YES in this field, but you have a plan or package with the bind option EXPLAIN(NO), then EXPLAIN processing does not occur during automatic rebind.

'NO' specifies that you do not want EXPLAIN processing to occur during the automatic rebind of a plan or package.

## 10. DPROP SUPPORT

|                    |                                    |
|--------------------|------------------------------------|
| Acceptable values: | 1, 2, 3                            |
| Default:           | 1                                  |
| Update:            | option 16 on panel DSNTIPB         |
| DSNZPxxx:          | DSN6SPRM EDPROP, DSN6SPRM<br>CHGDC |

Specify whether you want to use DataPropagator NonRelational to propagate SQL changes made to tables defined with DATA CAPTURE CHANGES.

'1' specifies that you do not intend to propagate changes.

'2' specifies that you intend to use DataPropagator NonRelational to propagate SQL changes, and that changes made to tables defined with DATA CAPTURE CHANGES are **only** allowed when the following conditions are met:

- Monitor trace class 6 is active
- DataPropagator NonRelational is installed
- The DB2 application is running in an IMS environment

If you choose 2 for DataPropagator NonRelational Support and monitor trace class 6 is not active, DataPropagator NonRelational is not installed, or the DataPropagator NonRelational application is not running in an IMS environment, then no changes to the DB2 table are permitted.

'3' specifies that data propagation will occur when the following conditions are met:

- Monitor trace class 6 is active
- DataPropagator NonRelational is installed
- The DB2 application is running in an IMS environment

The ANY option for DataPropagator NonRelational Support is intended for subsystems that need to propagate some data with DataPropagator NonRelational and need to propagate some data with a different program.

If you choose 3 for DataPropagator NonRelational support, an application that is not running in an IMS environment can update DB2 tables defined with DATA CAPTURE CHANGES. However, these changes are not propagated to IMS. You can protect your tables that should only be updated by DB2 applications running in an IMS environment by any of the following methods:

- Using the ENABLE parameter on BIND to specify a specific attachment facility through which updates to data propagation tables can be made.
- Defining a validation procedure for data propagation tables to allow only certain plans to update those tables.
- Using a group authorization ID to allow update authority for data propagation tables to a group of authorization IDs that can only run in the IMS environment.

See "Application Programming with IMS" on page 1-41 for more information about DataPropagator NonRelational.

## 11. SITE TYPE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | LOCALSITE, RECOVERYSITE    |
| Default:           | LOCALSITE                  |
| Update:            | option 16 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM SITETYP           |

Specify whether the current system is at a local site or a recovery site. LOCALSITE is defined as the site where the multiple image copies are made and are operational. RECOVERYSITE is defined as the site named as an alternative for recovery purposes.

The RECOVER utility looks at this value to determine what site the current system is on and recovers everything from the copies of data registered at that site. The RECOVER and MERGECOPY utilities look at this value to determine whether COPYDDN or RECOVERDDN is allowed with NEWCOPY NO.

## Application Programming Defaults Panel 1: DSNTIPF

The entries on this panel set application programming defaults. The values you specify on this panel are used as default values by the program preparation panels, the program preparation CLIST (DSNH), and the precompiler. They can also be used as defaults by other programs, such as Query Management Facility (QMF).

**Migrating or updating the parameters:** If you alter parameter values that specify that a change during migration or update is “not recommended,” it can make the syntax of existing SQL statements invalid or affect the way application programs run. Update is allowed, but must be handled with caution.

Values set here are contained in load module DSNHDECP, in library *prefix*.SDSNEXIT, which can be loaded and accessed by application programs. When modifying DSNHDECP, do so only by changing and running the installation CLIST.

Do not modify the data in DSNHDECP. If you modify any installation parameters by changing job DSNTIJUZ directly, then these values are not recorded for later updates, new installations, or migrations. Field names for the DSNHDECP macro are given with the corresponding parameters in this section.

For more information on the fields on this panel, see Section 4 of *Application Programming and SQL Guide*.

```
DSNTIPF INSTALL DB2 - APPLICATION PROGRAMMING DEFAULTS PANEL 1
====> _

Enter data below:

 1 LANGUAGE DEFAULT ====> IBMCOB ASM,C,CPP,COBOL,COB2,IBMCOB,FORTRAN,PLI
 2 DECIMAL POINT IS ====> . . or ,
 3 MINIMUM DIVIDE SCALE ====> NO NO or YES for a minimum of 3 digits
 to right of decimal after division
 4 STRING DELIMITER ====> DEFAULT DEFAULT, " or ' (COBOL or COB2 only)
 5 SQL STRING DELIMITER ====> DEFAULT DEFAULT, " or '
 6 DIST SQL STR DELIMTR ====> ' ' or "
 7 MIXED DATA ====> NO NO or YES for mixed DBCS data
 8 EBCDIC CODED CHAR SET====> 0 The CCSID of your SBCS or MIXED DATA
 9 ASCII CODED CHAR SET ====> 0 The CCSID of SBCS or mixed data. 0-65533.
10 DEF ENCODING SCHEME ====> EBCDIC EBCDIC or ASCII

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 44. Application Programming Defaults Panel: DSNTIPF

### 1. LANGUAGE DEFAULT

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | see below                  |
| Default:           | IBMCOB                     |
| Update:            | option 17 on panel DSNTIPB |
| DSNHDECP:          | DEFLANG                    |

Specify the default programming language for your site. Use any of the following:

|                                         |                                          |
|-----------------------------------------|------------------------------------------|
| <b>ASM</b> for High Level Assembler/MVS | <b>COB2</b> for VS COBOL II              |
| <b>C</b> for C Language                 | <b>IBMCOB</b> for IBM COBOL for MVS & VM |
| <b>CPP</b> for C++                      | <b>FORTRAN</b> for FORTRAN               |
| <b>COBOL</b> for OS/VS COBOL            | <b>PLI</b> for PL/I                      |

If you specify C or C++ in this field, you can *fold* SQL identifiers to upper case. However, this is not a default from any installation panel. For more information on this precompiler option see *Application Programming and SQL Guide* .

## 2. DECIMAL POINT IS

|                    |                                      |
|--------------------|--------------------------------------|
| Acceptable values: | . (period) or , (comma)              |
| Default:           | . (period)                           |
| Update:            | recommended only to recover an error |
| DSNHDECP:          | DECIMAL                              |

Specify whether the decimal point for numbers is the comma (,) or the period (.). Some nations customarily signify the number “one and one-half,” for instance, as “1.5”; other nations use “1,5” for the same value.

#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#

This parameter is used in the following cases:

- For dynamic SQL statements in packages or plans that are bound with DYNAMICRULES(RUN):
  - When the value of field DECIMAL POINT IS is COMMA, DB2 recognizes the comma as the decimal point for numbers.
  - When the value of field DECIMAL POINT IS is PERIOD, DB2 recognizes the period as the decimal point for numbers.
- For dynamic SQL statements in packages or plans that are bound with DYNAMICRULES(BIND):
  - If the value of field USE FOR DYNAMICRULES is NO, DB2 does not use the value in field DECIMAL POINT IS if you specify the option COMMA or PERIOD when you precompile the application that contains the dynamic SQL statements. DB2 uses the precompiler option to determine the decimal point for numbers.
  - If the value of field USE FOR DYNAMICRULES is YES, and the value of field DECIMAL POINT IS is PERIOD, DB2 recognizes the period as the decimal point for numbers.
  - If the value of field USE FOR DYNAMICRULES is YES, and the value of field DECIMAL POINT IS is COMMA, DB2 recognizes the comma as the decimal point for numbers.
- For static SQL statements in COBOL programs DECIMAL POINT IS specifies the default precompiler option (PERIOD or COMMA).

This parameter is the default for binds at this DB2 that are requested by a remote system that does not indicate whether the period or the comma is used to represent a decimal point. In most cases, however, requesting systems give DB2 this information.

### 3. MINIMUM DIVIDE SCALE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | YES, NO                    |
| Default:           | NO                         |
| Update:            | option 17 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM DECDIV3           |

Specify YES to retain at least three digits to the right of the decimal point after any decimal division. Certain accounting applications might need this option. Use NO, the default, to accept the usual rules for decimal division in SQL. For more information about decimal division in SQL, see Chapter 3 of *SQL Reference*.

### 4. STRING DELIMITER

|                    |                                             |
|--------------------|---------------------------------------------|
| Acceptable values: | DEFAULT, " (quotation mark), ' (apostrophe) |
| Default:           | DEFAULT                                     |
| Update:            | option 17 on panel DSNTIPB                  |
| DSNHDECP:          | DELIM                                       |

Specify the value of the string delimiter for COBOL. If you specify DEFAULT, the string delimiter is the quotation mark. This option is effective for all varieties of COBOL. See field 5 for a description of how to use this field with field 5 to get the desired set of character string delimiters for COBOL and SQL.

### 5. SQL STRING DELIMITER

|                    |                                             |
|--------------------|---------------------------------------------|
| Acceptable values: | DEFAULT, " (quotation mark), ' (apostrophe) |
| Default:           | DEFAULT                                     |
| Update:            | not recommended                             |
| DSNHDECP:          | SQLDELI                                     |

Specify the value of the SQL string delimiter that sets off character strings in dynamic SQL. This option is effective for all varieties of COBOL.

The value in this field also determines which character is the escape character for delimited identifiers in dynamic SQL. If you specify an apostrophe in this field, you get a quotation mark for your SQL escape character. If you specify a quotation mark in this field, you get an apostrophe for your SQL escape character.

For SQL embedded in COBOL programs, COBOL precompiler options specify which character is the SQL string delimiter and which character is the SQL escape character. If you specify DEFAULT in this field, a quote is passed to the precompiler as the default SQL string delimiter.

Some applications might require a particular value for the SQL STRING DELIMITER. Determine the required values for those applications before installing DB2.

Table 50 on page 2-167 shows you the different combinations of character string delimiters you get by specifying different values in fields 4 and 5.

| For this combination of character string delimiters... |             |              | Specify this in field 4 | Specify this in field 5 |
|--------------------------------------------------------|-------------|--------------|-------------------------|-------------------------|
| COBOL                                                  | Dynamic SQL | Embedded SQL |                         |                         |
| "                                                      | '           | "            | DEFAULT                 | DEFAULT                 |
| '                                                      | '           | '            | '                       | '                       |
| "                                                      | "           | "            | "                       | "                       |
| "                                                      | '           | '            | "                       | '                       |

The values you specify in fields 4 and 5 are also used by the program preparation panels, the DSNH CLIST, and the precompiler. Table 51 shows you why you might specify different combinations of values in these fields.

Table 51. Effect of Fields 4 and 5 on Precompiler Options

| Purpose                                                                                                                                                                                | Field 4 | Field 5 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------|
| Force APOST default (even in COBOL) and provide a default similar to APOST in SQL/DS                                                                                                   | '       | '       |
| Change dynamic query string delimiter to the quote. Helpful if you use COBOL with the quote option—allows queries to be tested with dynamic SQL and moved into the program more easily | "       | "       |
| Compatibility with the SQL/DS QUOTE option                                                                                                                                             | "       | '       |

## 6. DIST SQL STR DELIMTR

Acceptable values: ' (apostrophe) or " (quotation mark)  
 Default: ' (apostrophe)  
 Update: not recommended  
 DSNHDECP: DSQLDELI

Specify whether the apostrophe or the quotation mark is used as the SQL string delimiter for bind operations at this DB2 when the requester does not give DB2 that information. In most cases, requesters tell DB2 whether the apostrophe or the quotation mark is used as the SQL string delimiter

## 7. MIXED DATA

Acceptable values: YES, NO  
 Default: NO  
 Update: not recommended  
 DSNHDECP: MIXED

Specify whether the code points X'0E' and X'0F' have special meaning as the shift-out and shift-in controls for character strings that include double-byte characters.

- NO indicates that these code points have no special meaning. Therefore, all character strings are single-byte character set (SBCS) data.
- YES indicates that these code points have the special meaning described above. Therefore, character strings can be SBCS or MIXED data.

## 8. EBCDIC CODED CHAR SET

|                    |                                                           |
|--------------------|-----------------------------------------------------------|
| Acceptable values: | 0-65533                                                   |
| Default:           | 0 (no CCSID is used)                                      |
| Update:            | recommended only to recover an error                      |
| DSNHDECP:          | SCCSID (single-byte), MCCSID (mixed),<br>GCCSID (graphic) |

If your installation does not use character conversion, accept the default. You must specify a coded character set identifier (CCSID) if you specify any of the following values:

- AUTO or COMMAND for the DDF STARTUP OPTION field on panel DSNTIPR
- YES for the MIXED DATA field on panel DSNTIPF

To determine what single-byte CCSID to use, see Table 127 on page X-56.

To select a CCSID for mixed-data (an MCCSID), see Table 128 on page X-58. By specifying a CCSID you also receive system CCSIDs for your SBCS and GRAPHIC data.

### Attention:

- If MIXED DATA=YES, you must specify a Mixed Data CCSID from Table 128 on page X-58 or Table 129 on page X-59. An error occurs if you do not specify a CCSID or if the CCSID you specify is not listed in the table.
- If you specify a CCSID that is not a value in SYSSTRINGS.OUTCCSID, an error occurs when DB2 accesses SYSSTRINGS to determine if a conversion is provided. The CCSID is most likely incorrect and is probably not valid. If it is correct, conversions can be provided by adding rows to SYSSTRINGS as explained in “Adding a Character Conversion Procedure to DB2” on page X-63.
- If you specify a CCSID that appears as a value of SYSSTRINGS.OUTCCSID, but the CCSID is incorrect, data can be corrupted. For example, assume the coded character set used at your site is 37, but you specify 500 as the system CCSID. If DB2 receives data with a CCSID of 500, the data can be corrupted because character conversion does not occur. Conversely, if DB2 receives data with a CCSID other than 500 and a conversion is made from that CCSID to 500, the data can be corrupted because character conversion occurs. In both cases, the corruption is usually limited to special characters such as brackets and braces.
- If you specify an incorrect CCSID, you can correct it by altering the CODED CHAR SET field for your default encoding scheme. The alteration has no effect on bound SQL statements. Bound statements that refer to string variables include the CCSID of these variables. That CCSID is the system CCSID of the local DB2 at the time the statements were bound. Therefore, if the system CCSID is corrected *after* programs were bound, the change does not correct the CCSID of their input variables. The only way to correct the CCSID of these variables is to rebind the programs. This requirement is limited to programs that were bound to application servers other than the local DB2. There is no need to rebind local applications after you change the system CCSID because the bound form of their SQL statements indicates that character conversion does not occur.

If you specify either 930 or 5026, Katakana characters are allowed in ordinary identifiers and letters are not changed to upper case.



## 9. ASCII CODED CHAR SET

Acceptable values: 0-65533  
Default: 0 (no CCSID is used)  
Update: recommended only to recover an error  
DSNHDECP: ASCCSID (single-byte), AMCCSID (mixed),  
AGCCSID (graphic)

If your installation does not have any ASCII databases, table spaces, or tables, accept the default. You must specify a coded character set identifier (CCSID) for this field if you have ASCII databases, table spaces, or tables.

The recommended ASCII CCSIDs are in Table 52. See also the *Character Data Representation Architecture Reference* and *Character Data Representation Architecture Registry* for more information.

Table 52. Recommended ASCII CCSIDs

| Country        | ASCII CCSID CODES                                     |     |         |
|----------------|-------------------------------------------------------|-----|---------|
|                | PC                                                    | AIX | WINDOWS |
| United States  | 437                                                   | 819 | 1252    |
| Germany        | 850                                                   | 819 | 1252    |
| United Kingdom | 850                                                   | 819 | 1252    |
| Others         | See Table 127 on page X-56 and Table 129 on page X-59 |     |         |

To determine which single-byte ASCII CCSID to use, see Table 127 on page X-56.

## 10. DEF ENCODING SCHEME

Acceptable values: EBCDIC, ASCII  
Default: EBCDIC  
Update: not recommended  
DSNHDECP: ENScheme

Specify the format in which to store data in DB2. If you specify DEF ENCODING SCHEME=ASCII and MIXED DATA=YES, specify a mixed ASCII CCSID for ASCII CODED CHAR SET.

## Application Programming Defaults Panel 2: DSNTIP4

This panel is a continuation of DSNTIPF and is used to set application programming defaults. The values you specify on this panel are used as default values by the program preparation panels, the program preparation CLIST (DSNH), and the precompiler. They can also be used as defaults by other programs, such as Query Management Facility (QMF).

```

DSNTIP4 INSTALL DB2 - APPLICATION PROGRAMMING DEFAULTS PANEL 2
====> _

Enter data below:

 1 DATE FORMAT ====> ISO ISO, JIS, USA, EUR, LOCAL
 2 TIME FORMAT ====> ISO ISO, JIS, USA, EUR, LOCAL
 3 LOCAL DATE LENGTH ====> 0 10-254 or 0 for no exit
 4 LOCAL TIME LENGTH ====> 0 8-254 or 0 for no exit
 5 STD SQL LANGUAGE ====> NO NO or YES
 6 DECIMAL ARITHMETIC ====> DEC15 DEC15, DEC31, 15, 31
 7 CURRENT DEGREE ====> 1 1 or ANY
 8 CACHE DYNAMIC SQL ====> NO NO or YES

PRESS: ENTER to continue RETURN to exit HELP for more information

```

Figure 45. Application Programming Defaults Panel: DSNTIP4

### 1. DATE FORMAT

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | ISO, USA, EUR, JIS, LOCAL |
| Default:           | ISO                       |
| Update:            | not recommended           |
| DSNHDECP:          | DATE                      |

Specify one of the following abbreviations shown in Table 53 as a default output format to represent dates:

Table 53. Date Formats

| Format Name                                       | Abbreviation | Format      | Example    |
|---------------------------------------------------|--------------|-------------|------------|
| International Standards Organization              | ISO          | yyyy-mm-dd  | 1986-12-25 |
| IBM USA standard                                  | USA          | mm/dd/yyyy  | 12/25/1986 |
| IBM European standard                             | EUR          | dd.mm.yyyy  | 25.12.1986 |
| Japanese Industrial Standard Christian Era        | JIS          | yyyy-mm-dd  | 1986-12-25 |
| Locally defined (by an installation exit routine) | LOCAL        | your choice |            |

DB2 can accept a date in any format as input. It interprets the input date based on its punctuation and then provides date output in the format you specify for this

parameter. If you use LOCAL, you must provide a date exit routine to perform date formatting; for information, see Appendixes (Volume 2) of *Administration Guide*.

## 2. TIME FORMAT

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | ISO, USA, EUR, JIS, LOCAL |
| Default:           | ISO                       |
| Update:            | not recommended           |
| DSNHDECP:          | TIME                      |

Specify one of the following formats shown in Table 54 as a default output to represent times:

Table 54. Time Formats

| Format Name                                | Abbreviation | Format                  | Example               |
|--------------------------------------------|--------------|-------------------------|-----------------------|
| International Standards Organization       | ISO          | hh.mm.ss                | 13.30.05              |
| IBM USA standard                           | USA          | hh:mm AM<br>or<br>hh PM | 1:30 PM<br>or<br>1 PM |
| IBM European standard                      | EUR          | hh.mm.ss                | 13.30.05              |
| Japanese Industrial Standard Christian Era | JIS          | hh:mm:ss                | 13:30:05              |
| Locally defined (by exit routine)          | LOCAL        | your choice             |                       |

DB2 can accept a time in any format as input. It interprets the input time based upon its punctuation and then provides time output in the format you specify for this parameter. If you use LOCAL, you must provide a time exit routine to perform time formatting; for information, see Appendix B (Volume 2) of *Administration Guide*.

## 3. LOCAL DATE LENGTH

|                    |                 |
|--------------------|-----------------|
| Acceptable values: | 0, 10-254       |
| Default:           | 0               |
| Update or migrate: | not recommended |
| DSNHDECP:          | DATELEN         |

Accept the default value **0** if you want to use one of the IBM-supplied date formats (ISO, JIS, USA, or EUR). This indicates that no user-defined date format exists in your system. If you use a locally defined date exit routine, enter the length of the longest field required to hold a date. If you want your own date format to be the default, enter LOCAL for field 8 on this panel.

## 4. LOCAL TIME LENGTH

|                    |                 |
|--------------------|-----------------|
| Acceptable values: | 0, 8-254        |
| Default:           | 0               |
| Update or migrate: | not recommended |
| DSNHDECP:          | TIMELEN         |

Accept the default value **0** if you want to use one of the IBM-supplied time formats (ISO, JIS, USA, or EUR). This indicates that no user-defined time format exists in your system. If you use a locally defined time exit routine, enter the length of the

longest field required to hold a time. If you want your own time format to be the default, enter LOCAL for field 9 on this panel.

## 5. STD SQL LANGUAGE

|                    |                                                     |
|--------------------|-----------------------------------------------------|
| Acceptable values: | YES, NO                                             |
| Default:           | NO                                                  |
| Update:            | not recommended; cannot be changed during migration |
| DSNHDECP:          | STDSQL                                              |

To specify that the SQL language used in application programs conforms to the portions of the 1986 ANSI SQL standard implemented by DB2, choose YES.

If you choose NO, you specify that programs are written in accordance with the SQL language defined by DB2.

## 6. DECIMAL ARITHMETIC

|                    |                                                     |
|--------------------|-----------------------------------------------------|
| Acceptable values: | DEC15, DEC31, 15, 31                                |
| Default:           | DEC15                                               |
| Update:            | not recommended; cannot be changed during migration |
| DSNHDECP:          | DECARTH                                             |

Specify the rules to be used when both operands in a decimal operation have precisions of 15 or less. DEC15 specifies the rules which do not allow a precision greater than 15 digits, and DEC31 specifies the rules which allow a precision of up to 31 digits. The rules for DEC31 are always used if either operand has a precision greater than 15. If you chose DEC15 for your previous installation, choosing DEC31 can produce different results for operations on existing data.

This installation option applies to dynamic SQL and provides the default for the DEC precompiler option. DEC15 is sufficient for most sites. Do not choose DEC31 unless you are certain that you need the extra precision. If you use DEC31, you are more likely to get a bind error, particularly in division operations. See Chapter 3 of *SQL Reference* for information about arithmetic with two decimal operands.

## 7. CURRENT DEGREE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1, ANY                     |
| Default:           | 1                          |
| Update:            | option 18 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM CDSSRDEF          |

Specifies the default for the CURRENT DEGREE special register when no degree is explicitly set using the SQL statement SET CURRENT DEGREE. Accepting the default disables query parallelism.

## 8. CACHE DYNAMIC SQL

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | YES, NO                    |
| Default:           | NO                         |
| Update:            | option 18 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM CACHEDYN          |

# Specify whether to cache prepared, dynamic SQL statements for later use by  
# eligible application processes. These prepared statements will be cached in the  
# EDM pool. When specifying YES, you should consider this usage when calculating  
# your EDM pool size. See “ Calculating the EDM Pool Space for the Prepared  
# Statement Cache” on page 2-43 for storage estimating details.

## IRLM Panel 1: DSNTIPI

The entries on this panel affect the installation of the internal resource lock manager (IRLM).

You must use one IRLM for each DB2 subsystem. See "IRLM Address Space (IRLMPROC)" on page 2-35 for more information about DB2 and IRLM. See *Data Sharing: Planning and Administration* for recommendations about choosing values for fields on this panel.

**Updating the parameters:** The update option on each parameter indicates the correct procedure:

**Note A** Change by editing the IRLM start procedure.

**Note B** Change by editing the associated parameter in job DSNTIJUZ, the IRLM start procedure, and input member DSNTIDXA. Then, execute DSNTIJUZ and restart DB2.

```
DSNTIPI INSTALL DB2 - IRLM PANEL 1
===> _

Enter data below:

 1 INSTALL IRLM ===> YES IRLM is required for DB2. Should the
 IRLM distributed with DB2 be installed?
 2 SUBSYSTEM NAME ===> IRLM IRLM MVS subsystem name
 3 RESOURCE TIMEOUT ===> 60 Seconds to wait for unavailable resource
 4 AUTO START ===> YES Start IRLM if not up. YES or NO
 5 PROC NAME ===> IRLMPROC Name of start procedure for IRLM
 6 TIME TO AUTOSTART ===> 300 Time DB2 will wait for IRLM autostart
 7 UTILITY TIMEOUT ===> 6 Utility wait time multiplier
 8 U LOCK FOR RR/RS ===> NO Lock mode for update cursor with
 RR or RS isolation. YES or NO

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 46. IRLM Panel 1: DSNTIPI

### 1. INSTALL IRLM

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | YES, NO                  |
| Default:           | YES                      |
| Update:            | see note B on page 2-174 |
| DSNZPxxx:          | none                     |

Specify whether to provide IRLM subsystem entries in job DSNTIJMV and to build an IRLM procedure. If you specify NO, then no IRLM procedure is produced. On installation panel DSNTIPJ all values are ignored with the exception of fields 3 and 4.

If you specify YES, the required entries are provided and the IRLM procedure is built.

## 2. SUBSYSTEM NAME

|   |                    |                                                                                       |
|---|--------------------|---------------------------------------------------------------------------------------|
| # | Acceptable values: | 1-4 characters. First must be A-Z, #, \$, or @. Others must be A-Z, 1-9, #, \$, or @. |
| # | Default:           | IRLM                                                                                  |
|   | Update:            | see note B on page 2-174                                                              |
|   | DSNZPxxx:          | DSN6SPRM IRLMSID                                                                      |

Specify the name by which MVS knows the IRLM subsystem. The name is used for communication between DB2 and the IRLM. This name is included in the MVS subsystem table IEFSSNxx where xx is the value you supply in field 3 (SUBSYSTEM MEMBER) on installation panel DSNTIPM found on page 2-186.

If you installed the IRLM for IMS, DB2's IRLM name must be different. Two IRLMs residing in the same MVS system must have unique MVS subsystem names. If you already have IRLM Release 5 installed, use the MVS subsystem name for that IRLM. Otherwise, we recommend that you accept the default value, IRLM. For more information, see *IMS/ESA Operator's Reference*.

IRLM PROC parameter: IRLMNM

## 3. RESOURCE TIMEOUT

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | 1-3600                   |
| Default:           | 60                       |
| Update:            | see note B on page 2-174 |
| DSNZPxxx:          | DSN6SPRM IRLMRWT         |

|  
Specify the number of seconds before a time-out is detected. Time-out means that a lock request has waited for a resource (or for claims on a resource for a particular claim class to be released) longer than the number of seconds specified on this option. The value specified for this option must be an integer multiple of the DEADLOCK TIME on installation panel DSNTIPJ because IRLM uses its deadlock timer to initiate time-out detection, as well as deadlock detection. For data sharing, the actual timeout period is longer than the timeout value. See Chapter 7 of *Data Sharing: Planning and Administration* for an explanation of timeouts.

#  
#  
#  
For information about optimizing performance by managing DB2's use of locks, see Section 5 (Volume 2) of *Administration Guide*.

## 4. AUTO START

|                    |                   |
|--------------------|-------------------|
| Acceptable values: | YES, NO           |
| Default:           | YES               |
| Update:            | edit job DSNTIJUZ |
| DSNZPxxx:          | DSN6SPRM IRLMAUT  |

Specify whether DB2 automatically starts and stops the IRLM.

If you specify YES, then when DB2 starts, it tries to start the IRLM if the IRLM is not already started. When DB2 stops, it automatically stops the IRLM. We recommend YES if you use the IRLM only for a single DB2 system.

If you specify NO, DB2 terminates if the IRLM is not started when DB2 comes up.

## 5. PROC NAME

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | 1-8 characters           |
| Default:           | IRLMPROC                 |
| Update:            | see note B on page 2-174 |
| DSNZPxxx:          | DSN6SPRM IRLMPRC         |

Specify the name of the IRLM procedure that MVS invokes if field 4 is YES. The name *cannot* be the same as the subsystem name given for field 2.

The procedure is created during installation or migration by job DSNTIJMV and is placed in SYS1.PROCLIB. You can review it by examining DSNTIJMV.

## 6. TIME TO AUTOSTART

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | 1-3600                   |
| Default:           | 300                      |
| Update:            | see note B on page 2-174 |
| DSNZPxxx:          | DSN6SPRM IRLMSWT         |

Specify the IRLM wait time in seconds. This is the time that DB2 waits for the IRLM to start during autostart. If the time expires, DB2 abends.

## 7. UTILITY TIMEOUT

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1-254                      |
| Default:           | 6                          |
| Update:            | option 19 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM UTIMOUT           |

This option allows utilities to wait longer than SQL applications to access a resource.

Specify the number of resource time-out values (field 3) that a utility or utility command waits for a lock or for all claims on a resource of a particular claim class to be released. For example, if you use the default value, a utility can wait 6 times longer than an SQL application for a resource. For more information, see Section 5 (Volume 2) of *Administration Guide*.

## 8. U LOCK FOR RR/RS:

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | YES, NO                   |
| Default:           | NO                        |
| Update:            | option 19 on panel DSTIPB |
| DSNZPxxx:          | DSN6SPRM RRULOCK          |

Specify whether to use the U (UPDATE) lock when using repeatable read (RR) or read stability (RS) isolation to access a table. If you specify NO, the lock mode for operations with RR or RS is S (SHARE). If the cursor in your applications includes the clause FOR UPDATE OF, but updates are infrequent, S locks generally provide better performance.

If you specify YES, the lock mode for operations with RR or RS is U. If your applications make frequent updates with repeatable read isolation, the U lock might provide greater concurrency than the S lock. However, applications that require high concurrency are almost always more efficient if they use cursor stability (CS) isolation.



## IRLM Panel 2: DSNTIPJ

The entries on this panel affect several of the characteristics of IRLM time-sharing fields and other locking options. The default values are adequate for most sites under ordinary conditions. You must start DB2 and IRLM group names with a letter.

```
DSNTIPJ INSTALL DB2 - IRLM PANEL 2
===> _

Enter data below:

 1 CROSS MEMORY ===> NO Local storage and cross memory use
 2 MAXIMUM ECSA ===> 6M Control block storage (1M-999M)
 3 LOCKS PER TABLE(SPACE)===> 1000 Maximum before lock escalation
 4 LOCKS PER USER ===> 10000 Maximum before resource unavailable
 5 DEADLOCK TIME ===> 5 Detection interval in seconds

For DB2 data sharing ONLY enter data below:

 6 DEADLOCK CYCLE ===> 1 Number of LOCAL cycles before GLOBAL
 7 MEMBER IDENTIFIER ===> 1 Member ID for this IRLM (1-247)

 8 IRLM XCF GROUP NAME ===> DXRGROUP Name of IRLM XCF group
 9 LOCK ENTRY SIZE ===> 2 Initial allocation, in bytes (2,4,8)

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 47. IRLM Panel 2: DSNTIPJ

### 1. CROSS MEMORY

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | YES, NO                   |
| Default:           | NO                        |
| Update:            | edit IRLM start procedure |
| DSNZPxxx:          | none                      |

Specify whether the IRLM uses the cross-address-space program call. This parameter determines where the IRLM lock control block structure is stored. NO puts the IRLM lock control block structure in the extended common storage area (ECSA). This requires less processor time but can reduce the range of addresses available to private address spaces. With PC=NO, the MAXIMUM ECSA parameter is active and limits the amount of CSA and ECSA used by IRLM.

YES puts the lock control block structure in the IRLM private address space, and the program call instruction is used to address to the structure. With PC=YES, the MAXIMUM ECSA option is ignored. See "Common Service Area" on page 2-36 for more information about how DB2 uses this field.

IRLM PROC parameter: PC

## 2. MAXIMUM ECSA

# Acceptable values: 1M-999M  
| Default: 6M  
Update: edit IRLM start procedure  
DSNZPxxx: none

# Specify the maximum amount of common service area (CSA) and extended CSA  
# (ECSA) the IRLM for this DB2 uses for its lock control block structure. IRLM is not  
# prevented from using additional CSA and ECSA for other purposes. You can enter  
the value in bytes (such as 5242880), or use the abbreviations K for kilobytes and  
M for megabytes.

# Make sure you set this value high enough so that IRLM does not reach the limit.  
# IRLM only gets storage as it needs it, so it is better to specify a larger value than  
# you think you might need. You can also change the value dynamically by using the  
# MVS command MODIFY irlmproc,SET,CSA. See "Common Service Area" on  
# page 2-36 for more information about how IRLM uses storage.

IRLM PROC parameter: MAXCSA

## 3. LOCKS PER TABLE(SPACE)

Acceptable values: 0-50000  
Default: 1000  
Update: field 20 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM NUMLKTS

This value becomes the default value (SYSTEM) for the LOCKMAX clause of the  
SQL statements CREATE TABLESPACE and ALTER TABLESPACE.

For more information on how this parameter functions, see Section 5 (Volume 2) of  
*Administration Guide*.

## 4. LOCKS PER USER

Acceptable values: 0-100000  
Default: 10000  
Update: option 20 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM NUMLKUS

Specify the maximum number of page or row locks that a single application can  
hold concurrently on all table spaces. The maximum includes locks on data pages,  
index pages, subpages, and rows that the program acquires when it accesses table  
spaces. The limit applies to all table spaces defined with the LOCKSIZE PAGE,  
LOCKSIZE ROW, or LOCKSIZE ANY options. 0 means that there is no limit to the  
number of page and row locks a program can acquire.

DB2 assumes that 250 bytes of storage are required for each lock. If you specify  
NO for field 1, we recommend that you change the default value for this field to  
consider the available lock space. If you define referential constraints between  
tables, you might want a higher value for this field.

For more information, see Section 5 (Volume 2) of *Administration Guide*.

## 5. DEADLOCK TIME

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | 1-99                      |
| Default:           | 5                         |
| Update:            | edit IRLM start procedure |
| DSNZPxxx:          | none                      |

Specify the time, in seconds, of the local deadlock detection cycle. Depending on the value that you enter, IRLM might substitute a smaller maximum value. For IRLM V2.1, the maximum value used is 5 seconds. We recommend you do not exceed 5 seconds. The value specified for this field must be less than the value specified for RESOURCE TIMEOUT, field 3 on installation panel DSNTIPI. Otherwise, time-out detection supersedes deadlock detection. A deadlock is a situation where two or more requesters are waiting for resources held by the other. Deadlock detection is the procedure by which a deadlock and its participants are identified.

IRLM PROC parameter: DEADLOK

## 6. DEADLOCK CYCLE

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | 1-99                      |
| Default:           | 1                         |
| Update:            | edit IRLM start procedure |
| DSNZPxxx:          | none                      |

This option is used only for DB2 data sharing. We recommend you use the default value of 1 if you are using data sharing. If you do not use data sharing this field is ignored, but you still must supply a value. DEADLOCK CYCLE is the number of local deadlock cycles that must expire before the IRLM performs global deadlock detection processing.

IRLM PROC parameter: DEADLOK

## 7. MEMBER IDENTIFIER

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | 1-247                     |
| Default:           | 1                         |
| Update:            | edit IRLM start procedure |
| DSNZPxxx:          | none                      |

Specify an ID number that uniquely names this IRLM data sharing member within an IRLM group. We recommend that you correlate the IRLM member ID with the DB2 member name. For example, for DB2 member DSN1, specify an IRLM member ID of 1.

Note that this IRLM ID does not relate directly to the limit of IRLM members that can be in the data sharing group. That limit is determined by the current hardware limits (currently 32). If you edit the *irlmproc* directly, you can specify a value from 1 to 255. See *Command Reference* for the *irlmproc* command information.

This option is used only for DB2 data sharing.

IRLM PROC parameter: IRLMID

## 8. IRLM XCF GROUP NAME

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | 1-8 characters            |
| Default:           | DXRGROUP                  |
| Update:            | edit IRLM start procedure |
| DSNZPxxx:          | none                      |

Specify the name of the IRLM group. This name must be different from the DB2 group name. We recommend that this name begin with DXR. All members in the DB2 group must have the same IRLM XCF group name.

This option is used only for DB2 data sharing.

To avoid names that IBM uses for its MVS cross-system coupling facility (XCF) groups, the first character must be an upper-case letter J-Z unless the name begins with DSN. Do not use SYS as the first three characters, and do not use UNDESIG as the group name.

IRLM PROC parameter: GROUP

## 9. LOCK ENTRY SIZE

|                    |                           |
|--------------------|---------------------------|
| Acceptable values: | 2, 4, 8                   |
| Default:           | 2                         |
| Update:            | edit IRLM start procedure |
| DSNZPxxx:          | none                      |

Specify the initial size, in bytes, of individual lock entries in the lock table portion of the lock structure. To make the most efficient use of coupling lock structure space, use the default value.

DB2 converts the value for LOCK ENTRY SIZE to a corresponding value for the IRLM parameter MAXUSRS as shown in Table 55.

Table 55. Converting LOCK ENTRY SIZE to MAXUSRS values

| LOCK ENTRY SIZE | MAXUSRS value |
|-----------------|---------------|
| 2               | 7             |
| 4               | 23            |
| 8               | 32            |

## Protection Panel: DSNTIPP

The entries on this panel are used for security matters. For the password fields:

- If you are migrating, enter the passwords of your existing data sets.
- If you are defining a data set for the first time, you can enter a password for it in this panel.
- If a data set identified in a field already exists with no password, make the entry blank. In other words, if you are doing an update, you cannot change a password that you have previously specified.
- To use Resource Access Control Facility (RACF) protection and not passwords, make the password entries blank.
- If the data sets are managed by data facility storage management subsystem (DFSMS), the password does not apply; data sets defined to DFSMS should be protected by RACF or some similar external security system.

**Updating the parameters:** If you are migrating, DB2 for OS/390 Version 5 uses your DB2 for MVS/ESA Version 4 catalog, directory, work file databases, BSDS, active logs, and archive logs. Consequently, you cannot change the passwords for those objects when migrating.

Change passwords on panel DSNTIPP with the ALTER command of access method service. Then run the change log inventory utility, DSNJU003, to tell DB2 the new passwords.

Other entries can be changed by an update process after migration. See “Main Panel: DSNTIPA1” on page 2-99.

```
DSNTIPP INSTALL DB2 - PROTECTION
====> _

Enter data below:

 1 MVS CATALOG ==> DSNDEFPW MVS catalog control password
 2 BSDS PASSWORD ==> DSNOPER1 Bootstrap data sets password
 3 LOG PASSWORD ==> DBADMIN Active log data sets password
 4 ARCHIVE LOG PW ==> DBADMIN Archive log data sets password
 5 ARCHIVE LOG RACF ==> NO RACF protect archive log data sets
 6 DIRECTORY/CATALOG ==> DBADMIN DB2 directory and catalog password
 7 USE PROTECTION ==> YES DB2 authorization enabled. YES or NO
 8 SYSTEM ADMIN 1 ==> SYSADM Authid of system administrator
 9 SYSTEM ADMIN 2 ==> SYSADM Authid of system administrator
10 SYSTEM OPERATOR 1 ==> SYSOPR Authid of system operator
11 SYSTEM OPERATOR 2 ==> SYSOPR Authid of system operator
12 UNKNOWN AUTHID ==> IBMUSER Authid of default (unknown) user
13 RESOURCE AUTHID ==> SYSIBM Authid of Resource Limit Table creator
14 BIND NEW PACKAGE ==> BINDADD Authority required: BINDADD or BIND
15 PLAN AUTH CACHE ==> 1024 Size in bytes per plan (0 - 4096)
16 PACKAGE AUTH CACHE==> 32K Global - size in bytes (0-2M)

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 48. Protection Panel: DSNTIPP

## 1. MVS CATALOG

|                    |                |
|--------------------|----------------|
| Acceptable values: | 1-8 characters |
| Default:           | DSNDEFPW       |
| Update:            | see below      |
| DSNZPxxx:          | none           |

Specify the password required to add new entries to the DB2 integrated catalog facility catalog. If you use YES for DEFINE CATALOG (field 2 on installation panel DSNTIPA2), you can use this password in the catalog definition.

You cannot change this field during update. If you change this field during migration, job DSNTIJIN is the only place where the password is changed.

## 2. BSDS PASSWORD

|                    |                                      |
|--------------------|--------------------------------------|
| Acceptable values: | 1-8 characters                       |
| Default:           | DSNOOPER1                            |
| Update:            | see page 2-181; not during migration |
| DSNZPxxx:          | none                                 |

Specify the password used to define the bootstrap data sets. The operator must enter this password when prompted for it while starting DB2.

## 3. LOG PASSWORD

|                    |                                      |
|--------------------|--------------------------------------|
| Acceptable values: | 1-8 characters                       |
| Default:           | DBADMIN                              |
| Update:            | see page 2-181; not during migration |
| DSNZPxxx:          | none                                 |

Specify the password used to define and access the active log data sets. It is used when defining the active log VSAM clusters during DB2 initialization.

## 4. ARCHIVE LOG PW

|                    |                                      |
|--------------------|--------------------------------------|
| Acceptable values: | 1-8 characters                       |
| Default:           | DBADMIN                              |
| Update:            | see page 2-181; not during migration |
| DSNZPxxx:          | none                                 |

Specify the password used to define and access the archive log data sets. It is used when active log data sets are off-loaded to the archive log.

## 5. ARCHIVE LOG RACF

|                    |                                      |
|--------------------|--------------------------------------|
| Acceptable values: | YES, NO                              |
| Default:           | NO                                   |
| Update:            | see page 2-181; not during migration |
| DSNZPxxx:          | DSN6ARVP PROTECT                     |

Specify whether archive log data sets are to be protected with individual profiles with the Resource Access Control Facility (RACF) when they are created. If you use YES, RACF protection must be active for DB2. However, a value of YES also means that you cannot use RACF generic profiles for archive log data sets. In addition, RACF class TAPEVOL must be active if your archive log is on tape.

Otherwise, the off-load will fail. For information about using RACF, see Section 3 (Volume 1) of *Administration Guide*.

## 6. DIRECTORY/CATALOG

|                    |                |
|--------------------|----------------|
| Acceptable values: | 1-8 characters |
| Default:           | DBADMIN        |
| Update:            | see below      |
| DSNZPxxx:          | none           |

Specify the password for the DB2 catalog, directory, and temporary data sets. It is used during definition of VSAM clusters. It is kept in the BSDS and is used when the OPEN macro is issued.

This field cannot be changed during update. If this field is changed during migration, job DSNTIJIN is the only place where the password is changed. The directory and catalog passwords can be changed as noted on page 2-181.

## 7. USE PROTECTION

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | YES, NO                    |
| Default:           | YES                        |
| Update:            | option 21 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM AUTH              |

Specify whether DB2 will perform authorization checking. NO disables all authorization checking in DB2 and disables the GRANT statement (granting every privilege to PUBLIC); it is not recommended.

## 8. SYSTEM ADMIN 1

|                    |                                    |
|--------------------|------------------------------------|
| Acceptable values: | 1-8 characters, the first a letter |
| Default:           | SYSADM                             |
| Update:            | option 21 in panel DSNTIPB         |
| DSNZPxxx:          | DSN6SPRM SYSADM                    |

Specify the first of two authorization IDs with installation SYSADM authority. The two users with installation SYSADM authority are permitted access to DB2 in all cases. For the implications of this authority, and to understand REVOKE implications when changing this field, see Section 3 (Volume 1) of *Administration Guide*.

## 9. SYSTEM ADMIN 2

|                    |                                    |
|--------------------|------------------------------------|
| Acceptable values: | 1-8 characters, the first a letter |
| Default:           | SYSADM                             |
| Update:            | option 21 on panel DSNTIPB         |
| DSNZPxxx:          | DSN6SPRM SYSADM2                   |

Specify the second of two authorization IDs with installation SYSADM authority; see field 8.

If blanked, the value is set to the value of field 8.

## 10. SYSTEM OPERATOR 1

|                    |                                    |
|--------------------|------------------------------------|
| Acceptable values: | 1-8 characters, the first a letter |
| Default:           | SYSOPR                             |
| Update:            | option 21 on panel DSNTIPB         |
| DSNZPxxx:          | DSN6SPRM SYSOPR1                   |

Specify the first of two authorization IDs with installation SYSOPR authority. The two users with installation SYSOPR authority are permitted access to DB2 even if the DB2 catalog is unavailable. For the implications of this authority, see Section 3 (Volume 1) of *Administration Guide*.

If blanked, the value is set to the value of field 8.

## 11. SYSTEM OPERATOR 2

|                    |                                    |
|--------------------|------------------------------------|
| Acceptable values: | 1-8 characters, the first a letter |
| Default:           | SYSOPR                             |
| Update:            | option 21 on panel DSNTIPB         |
| DSNZPxxx:          | DSN6SPRM SYSOPR2                   |

Specify the second of two system operators with installation SYSOPR authority; see field 10.

If blanked, the value will be set to the value of field 10.

## 12. UNKNOWN AUTHID

|                    |                                    |
|--------------------|------------------------------------|
| Acceptable values: | 1-8 characters, the first a letter |
| Default:           | IBMUSER                            |
| Update:            | option 21 on panel DSNTIPB         |
| DSNZPxxx:          | DSN6SPRM DEFLTID                   |

Specify the authorization ID used if RACF is not available for batch access and USER= is not specified in the job statement. Null is not a valid value.

## 13. RESOURCE AUTHID

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 1-8 characters             |
| Default:           | SYSIBM                     |
| Update:            | option 21 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SYSP RLFAUTH           |

Specify the authorization ID used if you plan to use the resource limit facility (governor). Null is not a valid value.

## 14. BIND NEW PACKAGE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | BINDADD, BIND              |
| Default:           | BINDADD                    |
| Update:            | option 21 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM BINDNV            |

Specify whether BIND or BINDADD authority is required to BIND a new version of an existing package. If you accept the default, BINDADD, you allow only users with the BINDADD system privilege to create a new package. If you specify BIND, you allow users with the BIND privilege on a package or collection to create a new



version of an existing package when they bind it. You also allow users with PACKADM authority to add a new package or a new version of a package to a collection. See Section 3 (Volume 1) of *Administration Guide* for a full description of the privileges needed to bind a new package.

## 15. PLAN AUTH CACHE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 0-4096 in multiples of 256 |
| Default:           | 1024                       |
| Update:            | option 21 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM AUTHCACH          |

Specify the size of the authorization cache to be used if no CACHESIZE is specified on the BIND PLAN subcommand. Choose 0 if you do not want to use an authorization cache. For an authorization cache, you need 32 bytes of overhead + (8 bytes of storage × number of concurrent users). See Section 3 (Volume 1) of *Administration Guide* for more information about cache size.

## 16. PACKAGE AUTH CACHE

|                    |                            |
|--------------------|----------------------------|
| Acceptable values: | 0-2M                       |
| Default:           | 32K                        |
| Update:            | option 21 on panel DSNTIPB |
| DSNZPxxx:          | DSN6SPRM CACHEPAC          |

Specify how much storage to allocate for the caching of package authorization information for all packages on this DB2 member. 32K is enough storage for about 400 collection-ID.package-IDs. The cache is stored in the DSN1DBM1 address space.

## MVS PARMLIB Updates Panel: DSNTIPM

The entries on this panel produce the DSNTIJMV job that defines DB2 to MVS and updates the following PARMLIB members:

- IEFSSNxx, to define DB2 and IRLM as formal MVS subsystems
- IEAAPFxx, to authorize the *prefix*.SDSNLOAD, *prefix*.SDSNLINK, and *prefix*.SDSNEXIT libraries
- LNKLSTxx, to include the *prefix*.SDSNLINK library.

**Updating the parameters:** Different sites have different requirements for identifying DB2 to MVS; as a result, the updates that DSNTIJMV makes to MVS PARMLIB members might be incomplete. To ensure that the updates are complete, we recommend that you edit the MVS PARMLIB members directly when you install or migrate DB2. This is substantially easier than editing DSNTIJMV.

```
DSNTIPM INSTALL DB2 - MVS PARMLIB UPDATES
===>

Check data and reenter to change:

 1 SUBSYSTEM NAME ===> DSN1 Name for connecting to DB2
 2 COMMAND PREFIX ===> -DSN1 DB2 subsystem command prefix
 3 SUBSYSTEM MEMBER ===> 00 xx in IEFSSNxx
 4 SUBSYSTEM SEQUENCE ===> 88888888 Sequence number for insertion
 5 AUTH MEMBER ===> 00 xx in IEAAPFxx APF member name
 6 AUTH SEQUENCE ===> 88888888 Sequence number for insertion
 7 LINK LIST ENTRY ===> 00 xx in LNKLSTxx for DSNLINK
 8 LINK LIST SEQUENCE ===> 88888888 Sequence number for insertion
 9 COMMAND SCOPE ===> STARTED SYSTEM, SYSPLEX, or STARTED

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 49. MVS PARMLIB Updates Panel: DSNTIPM

### 1. SUBSYSTEM NAME

|                    |                                                                                       |
|--------------------|---------------------------------------------------------------------------------------|
| Acceptable values: | 1-4 characters. First must be A-Z, #, \$, or @. Others must be A-Z, 1-9, #, \$, or @. |
| Default:           | DSN1                                                                                  |
| Update:            | see above                                                                             |
| DSNZPxxx:          | none                                                                                  |

Specify the MVS subsystem name for DB2. The name is used in member IEFSSNxx of SYS1.PARMLIB.

## 2. COMMAND PREFIX

|                    |                                                                 |
|--------------------|-----------------------------------------------------------------|
| Acceptable values: | 1-8 characters. First one must be a non-alphanumeric character. |
| Default:           | -DSN1 (hyphen concatenated with subsystem name)                 |
| Update:            | see 2-186                                                       |
| DSNZPxxx:          | none                                                            |

Specify the DB2 command prefix. When the prefix appears at the beginning of a command entered at an MVS operator's console, MVS passes the command to DB2 for processing. The command prefix is used in the DB2 entry of member IEFSSNxx of SYS1.PARMLIB.

The first character of the command prefix must be a character in Table 56. The remaining characters of the command prefix must be from Table 56, A-Z, or 0-9.

Table 56. Allowable Special Characters for the Command Prefix

| Name                    | Character | Hexadecimal Representation |
|-------------------------|-----------|----------------------------|
| cent sign               | ¢         | X'4A'                      |
| period                  | .         | X'4B'                      |
| less than sign          | <         | X'4C'                      |
| plus sign               | +         | X'4E'                      |
| vertical bar            |           | X'4F'                      |
| ampersand <sup>1</sup>  | &         | X'50'                      |
| exclamation point       | !         | X'5A'                      |
| dollar sign             | \$        | X'5B'                      |
| asterisk                | *         | X'5C'                      |
| right parenthesis       | )         | X'5D'                      |
| semi-colon              | ;         | X'5E'                      |
| hyphen                  | -         | X'60'                      |
| slash                   | /         | X'61'                      |
| percent sign            | %         | X'6C'                      |
| underscore              | _         | X'6D'                      |
| question mark           | ?         | X'6F'                      |
| colon                   | :         | X'7A'                      |
| number sign             | #         | X'7B'                      |
| at sign                 | @         | X'7C'                      |
| apostrophe <sup>2</sup> | '         | X'7D'                      |
| equal sign              | =         | X'7E'                      |
| quotation marks         | "         | X'7F'                      |

1. To use the ampersand (&), accept the default in this field, then edit job DSNTIJMV and specify the ampersand as the command prefix.
2. To use the apostrophe ('), you must code two consecutive apostrophes in your IEFSSNxx member. For example, the entry for subsystem DB2A with a command prefix of 'DB2A and a scope of started looks like this:  
DB2A,DSN3INI,' 'DSN3EPX,' 'DB2A,'

# 3. To use the equal sign (=), accept the default command prefix, then edit job  
# DSNTIJMV and replace the dash (–) with the equal sign as the first character  
# of the command prefix.

Do not use the JES2 backspace character as a command prefix character. Do not assign a command prefix that is used by another subsystem or that can be interpreted as belonging to more than one subsystem or MVS application. Specifically, do not specify a multiple-character command prefix that is a subset or a superset of another command prefix beginning from the first character. For example, it is invalid to assign '-' to one subsystem and '-DB2A' to another. Similarly, it is also invalid to assign '?DB2' to one subsystem and '?DB2A' to another. It is valid to assign '-DB2A' and '-DB2B' to different DB2 subsystems.

| To use multiple-character command prefixes, have the system programmer update  
# the IEFSSNxx subsystem definition statements in SYS1.PARMLIB. For more  
# information about the SYS1.PARMLIB IEFSSNxx statement see “DSNTIJMV  
# Updates to SYS1.PARMLIB” on page 2-234.

### 3. SUBSYSTEM MEMBER

Acceptable values: 2 alphanumeric characters  
Default: 00  
Update: see page 2-186  
DSNZPxxx: none

Specify the last two characters (xx) of the name of member IEFSSNxx of SYS1.PARMLIB. The subsystem member name indicates the available MVS subsystems, including DB2 and the IRLM.

### 4. SUBSYSTEM SEQUENCE

Acceptable values: 1-99999995  
Default: 88888888  
Update: see page 2-186  
DSNZPxxx: none

Specify any number greater than the highest sequence number already used in the IEFSSNxx PARMLIB member.

### 5. AUTH MEMBER

Acceptable values: 2 alphanumeric characters  
Default: 00  
Update: see page 2-186  
DSNZPxxx: none

Specify the last two characters (xx) of the name of member IEAAPFxx of SYS1.PARMLIB. This member is used for authorized program facility (APF) authorization of the *prefix*.SDSNLOAD, *prefix*.SDSNLINK, and *prefix*.SDSNEXIT libraries. This data set must be APF-authorized. The member name must currently exist for the MVS update job DSNTIJMV to function correctly.

If you are using MVS/ESA Version 4 Release 3, you might use the PROGxx member instead of the IEAAPFxx member. In this case, you must manually name the PROGxx member—job DSNTIJMV does not do it for you.

## 6. AUTH SEQUENCE

Acceptable values: 1-99999995  
Default: 88888888  
Update: see page 2-186  
DSNZPxxx: none

Specify any number greater than the highest sequence number already used in the IEAAPFxx PARMLIB member.

## 7. LINK LIST ENTRY

Acceptable values: 2 alphanumeric characters  
Default: 00  
Update: see page 2-186  
DSNZPxxx: none

Specify the last two characters of LNKLSTxx as needed to include the *prefix.SDSNLINK* library.

## 8. LINK LIST SEQUENCE

Acceptable values: 1-99999999  
Default: 88888888  
Update: see page 2-186  
DSNZPxxx: none

Specify any number greater than the highest sequence number already used in the LNKLSTxx PARMLIB member.

## 9. COMMAND SCOPE

Acceptable values: SYSTEM, SYSPLEX, STARTED  
Default: STARTED  
DSNZPxxx: none

Specify the scope of the command prefix.

**SYSTEM** The scope of commands is for one MVS system. The command prefix is registered at MVS IPL.

**SYSPLEX** The scope of commands is for the entire Sysplex. The command prefix is registered at MVS IPL.

**STARTED** The scope of commands is for the entire Sysplex. The command prefix is registered at DB2 startup, and deregistered when DB2 stops.

Although STARTED specifies a Sysplex scope, it can be used for a DB2 in a non-data-sharing environment as well. We recommend STARTED if you intend to use MVS's automatic restart manager, or if you might move this DB2 into a data sharing group.

## Active Log Data Set Parameters: DSNTIPL

The entries on this panel define characteristics of active log data sets.

**Performance note:** Several fields on this panel affect DB2's use of logging. Extra consideration should be taken when determining the values associated with fields on this panel. For a description of DB2 logging, see Section 4 (Volume 1) of *Administration Guide*.

```
DSNTIPL INSTALL DB2 - ACTIVE LOG DATA SET PARAMETERS
===> _

Enter data below:

 1 NUMBER OF LOGS ===> 3 Number data sets per active log copy (2-31)
 2 INPUT BUFFER ===> 28K Size in bytes (28K-60K)
 3 OUTPUT BUFFER ===> 400K Size in bytes (40K-4000K)
 4 WRITE THRESHOLD ===> 20 Buffers filled before write (1-256)
 5 ARCHIVE LOG FREQ ===> 24 Hours per archive run
 6 UPDATE RATE ===> 3600 Updates, inserts, and deletes per hour

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 50. Log Data Sets Panel: DSNTIPL

### 1. NUMBER OF LOGS

Acceptable values: 2-31  
Default: 3  
Update: cannot change during update or migration  
DSNZPxxx: none

# Specify the number of data sets for each copy of the active log. If you used the  
# DSNJU003 utility to modify the number of logs, your modified number is not  
# reflected on this panel during migration or update. See "Updating Other  
# Parameters" on page 2-230 for details.

### 2. INPUT BUFFER

Acceptable values: 28K-60K  
Default: 28K  
Update: option 23 on panel DSNTIPB  
DSNZPxxx: DSN6LOGP INBUFF

Specify the size of the input buffer used for reading active and archive log data sets. Each job requesting information from the log will have an input buffer dedicated to it. Each time a read operation is started during use of the RECOVER utility or during DB2 restart, a buffer of the specified size is allocated. You can

enter the value in bytes (for example, 32768) or use the abbreviation K for kilobytes (for example, 32K).

A larger buffer size generally favors the speed of reading the active log.

### 3. OUTPUT BUFFER

Acceptable values: 40K-4000K  
Default: 400K  
Update: option 23 on panel DSNTIPB  
DSNZPxxx: DSN6LOGP OUTBUFF

Specify the size of the output buffer used for writing active log data sets. You can enter the value in bytes (for example, 40960) or use the abbreviation K for kilobytes (for example, 40K). The larger the output buffer, the more likely a requested RBA can be found without a read request.

### 4. WRITE THRESHOLD

Acceptable values: 1-256  
Default: 20  
Update: option 23 on panel DSNTIPB  
DSNZPxxx: DSN6LOGP WRTHRSH

Specify the number of buffer pages to be filled before starting a write.

The larger the WRITE THRESHOLD value, the less often the contents of the output buffer are written to the active log. We recommend not exceeding 20 percent of the total number of buffer pages so that DB2 will not have to wait for an available buffer.

### 5. ARCHIVE LOG FREQ

Acceptable values: 1-200  
Default: 24  
Update: cannot change during update  
DSNZPxxx: none

Estimate the interval in hours at which the active log is off-loaded to the archive log. If you accept the default value of 24, the active log is off-loaded about once each day.

### 6. UPDATE RATE

Acceptable values: 1-16M  
Default: 3600  
Update: cannot change during update  
DSNZPxxx: none

Estimate the number of inserts, updates, and deletes expected per hour in your subsystem.

The size calculations in the DSNTINST CLIST assume that about 400 bytes of data are logged for each insert, update, and delete. The amount of data logged for these changes might be different at your site. Therefore, consider changing the size of the log data sets after you gain some experience with DB2 and have a better idea of how many bytes of data are logged for each change. Generally, if you have a

| subsystem that is tuned for maximum efficiency, you can expect to log about ten  
| gigabytes of data per hour while processing several millions of updates and inserts.

Together, the UPDATE RATE and the ARCHIVE LOG FREQ (field 5) determine the size of the active logs.



## Archive Log Data Set Parameters: DSNTIPA

The entries on this panel define the characteristics of archive log data sets.

**Updating the parameters:** All the parameters on this panel can be updated by their subsystem parameter name.

```
DSNTIPA INSTALL DB2 - ARCHIVE LOG DATA SET PARAMETERS
====>
Enter data below:

 1 ALLOCATION UNITS ==> BLK Blk, Trk, or Cyl
 2 PRIMARY QUANTITY ==> Primary space allocation
 3 SECONDARY QTY. ==> Secondary space allocation
 4 CATALOG DATA ==> NO YES or NO to catalog archive data sets
 5 DEVICE TYPE 1 ==> TAPE Unit name for COPY1 archive logs
 6 DEVICE TYPE 2 ==> Unit name for COPY2 archive logs
 7 BLOCK SIZE ==> 28672 Rounded up to 4096 multiple
 8 READ TAPE UNITS ==> 2 Maximum allocated read tape units
 9 DEALLOC PERIOD ==> 0 Time interval to deallocate tape units
10 RECORDING MAX ==> 1000 Number of data sets recorded in BSDS
11 WRITE TO OPER ==> YES Issue WTOR before mount for archive
12 WTOR ROUTE CODE ==> 1,3,4 Routing codes for archive WTORs

13 RETENTION PERIOD ==> 9999 Days to retain archive log data sets
14 QUIESCE PERIOD ==> 5 Maximum quiesce interval (1-999)
15 COMPACT DATA ==> NO YES or NO for data compaction

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 51. Archive Log Data Sets Panel: DSNTIPA

### 1. ALLOCATION UNITS

Acceptable values: Blk, Trk, or Cyl  
Default: Blk  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP ALCUNIT

#

Specify the units in which primary and secondary space allocations are obtained.

### 2. PRIMARY QUANTITY

Acceptable values: 1-99999999  
Default: blank  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP PRIQTY

#

Specify the primary space allocation for a DASD data set, in units of ALCUNIT. If you use the default, the CLIST calculates this space using block size and size of the log.

### 3. SECONDARY QTY.

# Acceptable values: 1-9999999  
Default: blank  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP SECQTY

Specify the secondary space allocation for a DASD data set, in units of ALCUNIT. If you use the default, the CLIST calculates this space using block size and size of the log.

### 4. CATALOG DATA

# Acceptable values: YES, NO  
Default: NO  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP CATALOG

| Specify whether archive log data sets are to be cataloged in the primary integrated catalog facility catalog. This option is only meaningful if you specify tape for the DEVICE TYPE 1 or DEVICE TYPE 2 fields on this panel, because DB2 requires that all archive log data sets allocated on DASD be cataloged. If you choose to archive to DASD, then the catalog option must be set to YES. If the catalog option is set to NO and you decide to place your archive log data sets on DASD, you receive message DSNJ072E each time an archive log data set is allocated, and the DB2 subsystem catalogs the data set.

### 5. DEVICE TYPE 1

# Acceptable values: device type or unit name  
Default: TAPE  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP UNIT

Specify the device type or unit name for storing archive log data sets. The value can be any alphanumeric string. If you choose to archive to DASD, you can specify a generic device type with a limited volume range. DB2 requires that all archive log data sets allocated on DASD be cataloged. If the device type specifies DASD, then set field 4 (CATALOG DATA) to YES.

If the unit name specifies DASD, the archive log data sets cannot extend to another volume. If the unit name specifies a tape device, DB2 can extend to a maximum of 20 volumes. If you chose to use DASD, make the primary space allocation (field 2) large enough to contain all of the data coming from the active log data sets.

### 6. DEVICE TYPE 2

# Acceptable values: device type or unit name  
Default: none  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP UNIT2

Specify the device type or unit name for storing the second copy of archive log data sets (COPY2 data sets), as for field 5.

## 7. BLOCK SIZE

Acceptable values: 8192-28672  
Default: 28672  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP BLKSIZE

Specify the block size of the archive log data set. The block size must be compatible with the device type you use for archive logs. The value is rounded up to the next multiple of 4096 bytes. You can also enter the value with a K; for example, 28K.

If the archive log is written to tape, using the largest possible block size improves the speed of reading the archive logs. Use table Table 57 as a guide.

Table 57. Recommended Block Size Values

| Archive log device | Block size |
|--------------------|------------|
| Tape               | 28672      |
| 3380               | 20480      |
| 3390 or RAMAC      | 24576      |

## 8. READ TAPE UNITS

Acceptable values: 1-99  
Default: 2  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6LOGP MAXRTU

Specify the maximum number of dedicated tape units that can be allocated to read archive log tape volumes concurrently. This installation option, along with DEALLOC PERIOD, allows DB2 to optimize archive log reading from tape devices.

*In a data sharing environment* the archive tape is not available to other members of the group until the deallocation period expires. You may not want to use this option in a data sharing environment unless all recover jobs are submitted from the same member.

We recommend that you set the READ TAPE UNITS value to be at least one less than the number of tape units available to DB2. If you do otherwise, the OFFLOAD process could be delayed, which would affect the performance of your DB2 subsystem. For maximum through-put during archive log processing, specify the largest value possible for this option keeping in mind that you need at least one tape unit for OFFLOAD processing. You can override this value by using the SET ARCHIVE command.

## 9. DEALLOC PERIOD

Acceptable values: see below  
Default: 0  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6LOGP DEALLCT

Specify the length of time an archive read tape unit is allowed to remain unused before it is deallocated. Use *minutes*, *seconds* (blank or 1-1439, blank or 0-59), 1440 (minutes), or NOLIMIT. Specifying NOLIMIT allows maximum optimization opportunities.

When archive log data is being read from tape, we recommend that you set this value high enough to allow DB2 to optimize tape handling for multiple read applications. When all tape reading is complete, you can update this option with the SET ARCHIVE command.

#### 10. RECORDING MAX

Acceptable values: 10-1000  
Default: 1000  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6LOGP MAXARCH

#

Specify the maximum number of archive log volumes to be recorded in the BSDS. When this number is exceeded, recording resumes at the beginning of the BSDS.

You must create image copies of all DB2 objects, probably several times, before the archive log data sets are discarded. If you fail to retain an adequate number of archive log data sets for all the image copies, you might have to cold start or reinstall DB2. In both cases, data is lost.

For information about managing the log and determining how long to keep the logs, refer to Section 4 (Volume 1) of *Administration Guide*.

#### 11. WRITE TO OPER

Acceptable values: YES, NO  
Default: YES  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP ARCWTOR

#

Specify whether to send a message to the operator and wait for an answer before attempting to mount an archive log data set. Other DB2 users can be forced to wait while the mount is pending. They are not affected while DB2 is waiting for a response to the message.

Specify NO if you use a device, such as DASD, that does not have long delays for mounts. Specify YES if you use a device for storing archive log data sets, such as tape, that requires long delays for mounts. Field 5 (DEVICE TYPE 1) specifies the device type or unit name.

#### 12. WTOR ROUTE CODE

Acceptable values: see below  
Default: 1,3,4  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP ARCWRTC

#

Specify the list of route codes from messages from the archive log data sets to the operator. You can specify from 1 to 16 route codes. Separate numbers in the list by commas only, not by blanks.

For descriptions of the routing codes, see *MVS/ESA System Codes*. The routing codes are also discussed in the description of the WTO macro in *MVS/ESA Programming: Authorized Assembler Services Reference, Volumes 1-4*.

|  
|  
|

### 13. RETENTION PERIOD

Acceptable values: 0-9999  
Default: 9999  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP ARCRETN

#

Specify the number of days that DB2 is to retain archive log data sets. The retention period is often used in tape management systems to control the reuse and scratching of data sets and tapes. DB2 uses the value as the value for the dynamic allocation parameter DALRETPD when archive log data sets are created.

The retention period set by DFSMSdfp's storage management subsystem (SMS) can be overridden by this DB2 parameter. Typically, the retention period is set to the smaller value specified by either DB2 or SMS. The storage administrator and database administrator should agree on a retention period value that is appropriate for DB2.

The retention period is added to the current date to calculate the expiration date.

#### Attention:

Due to the wide variety of tape management systems and the opportunity for external manual overrides of retention periods, DB2 does not have an automated method to delete the archive log data sets from the BSDS inventory of archive log data sets. Therefore, the information about an archive log data set might be in the BSDS long after it has been scratched by a tape management system after its retention period expired. Conversely, the maximum number of archive log data sets can have been exceeded (see field 8), and the data from the BSDS dropped long before the data set has reached its expiration date.

For information about the archive log data sets and how they can be managed using the change log inventory utility (DSNJU003), refer to Section 4 (Volume 1) of *Administration Guide*.

### 14. QUIESCE PERIOD

Acceptable values: 0-999  
Default: 5  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP QUIESCE

#

Specify the maximum amount of time (in seconds) DB2 is allowed to attempt a full system quiesce.

This parameter requires some tuning. If you specify too short an interval, the quiesce period expires before a full quiesce is accomplished. If you specify too long an interval, the quiesce period could cause unnecessary DB2 lock contention and time-outs. For more information on the quiesce mode of the Archive Log command, see Section 4 (Volume 1) of *Administration Guide*.

#

### 15. COMPACT DATA

#

Acceptable values: YES or NO  
Default: NO  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP COMPACT

Specify whether data written to archive logs should be compacted. This option only applies to data written to a 3480 device that has the improved data recording capability (IDRC) feature. When this feature is turned on, hardware in the tape control unit writes data at a much higher density than normal, allowing for more data on a volume. Specify NO if you do not use a 3480 device with the IDRC feature or a 3490 base model, with the exception of the 3490E. Specify YES if you want the data to be compacted.

|  
|  
|  
|  
|  
|  
|  
|  
|

If you use compression or auto-blocking on the tape unit, you will need to ensure that you do not read backwards on the tape unit. You can do this by increasing the size and number of active log data sets and by monitoring long running units of recovery with the UR CHECK FREQ (panel DSNTIPN) or another monitor. The alternative to monitoring the units of work and increasing active log space is archiving to disk and then using another facility, such as DFSMSHsm to archive the archive log from disk to tape. Be aware that data compressed to tape can only be read using a device that supports the IDRC feature. This could be a concern when you send archive tapes to another site for remote recovery.

## Databases and Spaces to Start Automatically Panel: DSNTIPS

The entries on this panel name the databases, table spaces, and index spaces to restart automatically when you start DB2.

**Updating the parameters:** All parameters on this panel can be updated using their subsystem parameter name.

```
DSNTIPS INSTALL DB2 - DATABASES AND SPACES TO START AUTOMATICALLY
===> _

Enter data below:

1 ===> RESTART RESTART or DEFER the objects named below.
 The objects to restart or defer can be ALL in item 2, a database
 name, or database name.space name.

2 ==> ALL 14 ==>
3 ==> 15 ==>
4 ==> 16 ==>
5 ==> 17 ==>
6 ==> 18 ==>
7 ==> 19 ==>
8 ==> 20 ==>
9 ==> 21 ==>
10 ==> 22 ==>
11 ==> 23 ==>
12 ==> 24 ==>
13 ==> 25 ==>
 26 ==>
 27 ==>
 28 ==>
 29 ==>
 30 ==>
 31 ==>
 32 ==>
 33 ==>
 34 ==>
 35 ==>
 36 ==>
 37 ==>

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 52. Databases and Spaces to Start Automatically Panel: DSNTIPS

### 1. RESTART OR DEFER

Acceptable values: RESTART, DEFER  
Default: RESTART  
Update: option 25 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RESTART

#

Specify whether DB2 will restart or defer processing for the objects listed in fields 2 through 37 when DB2 is started. RESTART causes DB2 to perform restart processing for the objects listed. DEFER causes DB2 not to perform restart processing for the objects.

### 2 - 37. START NAMES

Acceptable values: ALL, space names  
Default: ALL  
Update: option 25 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM ALL

#

Specify the names of the databases, table spaces, and index spaces for which you want to control restart processing. Enter one of the following values for these fields:

- **ALL** on field 2 (leaving fields 3 - 37 blank) to restart or defer all DB2 databases and spaces. This is the default.

DEFER ALL defers recovery of all objects, including DB2 catalog objects. For information about the restart process and the implications of deferring objects at restart time, see Section 4 (Volume 1) of *Administration Guide*.

- Database name to restart or defer all spaces in that database.
- Table space or index space name in the format “database-name.space-name” to restart or defer the individual table or index space.

You can specify up to 36 object names on this panel. If you want to control restart processing for more than 36 objects, edit job DSNTIJUZ after you run the CLIST and add the object names as ending positional parameters to macro DSN6SPRM. You can add up to 2500 object names in DSNTIJUZ.



## Distributed Data Facility: DSNTIPR

The entries on this panel control the starting of the distributed data facility (DDF) and specify names used to connect another DB2 subsystem.

To use DDF, you must have VTAM installed, even if you are using only TCP/IP connections. If you do not have VTAM installed, see “Installing a Communications Network” on page 2-257 for instructions on installing VTAM.

```
DSNTIPR INSTALL DB2 - DISTRIBUTED DATA FACILITY PANEL 1
===> _

DSNT512I WARNING: ENTER UNIQUE NAMES FOR LUNAME AND LOCATION NAME
Enter data below:

 1 DDF STARTUP OPTION ===> NO NO, AUTO, or COMMAND
 2 DB2 LOCATION NAME ===> LOC1 The name other systems use to
 refer to this DB2
 3 DB2 NETWORK LUNAME ===> LU1 The name VTAM uses to refer to this DB2
 4 DB2 NETWORK PASSWORD ===> Password for DB2's VTAM application
 5 RLST ACCESS ERROR ===> NOLIMIT NOLIMIT, NORUN, or 1-5000000
 6 RESYNC INTERVAL ===> 2 Minutes between resynchronization period
 7 DDF THREADS ===> ACTIVE (ACTIVE or INACTIVE) Status of a
 database access thread that commits
 or rolls back and holds no database
 locks or cursors.

 8 DB2 GENERIC LUNAME ===> Generic VTAM LU name for this DB2
 subsystem or data sharing group.
 9 IDLE THREAD TIMEOUT ===> 0 0 or seconds until dormant server ACTIVE
 thread will be terminated

10 EXTENDED SECURITY ===> NO Allow change password and descriptive
 security error codes? YES or NO.

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 53. Distributed Data Facility Panel: DSNTIPR

### 1. DDF STARTUP OPTION

Acceptable values: NO, AUTO, COMMAND  
Default: NO  
Update: option 26 on panel DSNTIPB  
DSNZPxxx: DSN6FAC DDF

Specify whether or not to load DDF, and if DDF is loaded, how to start it.

NO signifies that you do not want the DDF loaded at DB2 startup and that it cannot be started with a command. If you specify NO, the remaining fields on this panel are ignored and the stored procedures sample application and DDF jobs (DNSTEJ6S, DSNTJ6P, DSNTJ6, DSNTJ6D, and DSNTJ6T) are not edited.

AUTO specifies that this facility is automatically initialized and started when the DB2 subsystem is started. The DDF address space is started as part of DDF initialization.

COMMAND specifies that the facility is initialized at DB2 startup and is prepared to receive the -DSN1 START DDF command. The DDF address space is started as part of DDF initialization.

If AUTO or COMMAND is specified, the remaining fields are mandatory. The repository for the field names (LOCATION, LUNAME, and PASSWORD) is the bootstrap data set (BSDS).

## 2. DB2 LOCATION NAME

Acceptable values: 1-16 alphanumeric characters  
Default: LOC1  
Update: see Update on page 2-230  
DSNZPxxx: none

Specify the unique name which application requesters use to connect to this DB2 subsystem. The name must begin with a letter and must not contain special characters. Acceptable characters are A-Z, 0-9, and underscore.

You must specify a value, even if you do not use DDF .

## 3. DB2 NETWORK LUNAME

Acceptable values: 1-8 alphanumeric characters  
Default: LU1  
Update: see Update on page 2-230  
DSNZPxxx: none

Specify the logical unit name (LU name) for this DB2 subsystem. This name uniquely identifies this DB2 subsystem to VTAM. It is also used to uniquely identify logical units of work within DB2 trace records. The name must begin with a letter and must not contain special characters.

You must specify a value.

## 4. DB2 NETWORK PASSWORD

Acceptable values: 1-8 alphanumeric characters  
Default: none  
Update: see Update on page 2-230  
DSNZPxxx: none

If provided, this field specifies the password used by VTAM to recognize this DB2 subsystem. This password must then be supplied to VTAM on the VTAM APPL definition statement. The password must begin with a letter and must not contain special characters.

## 5. RLST ACCESS ERROR

Acceptable values: NOLIMIT, NORUN, 1-5000000  
Default: NOLIMIT  
Update: option 26 on panel DSNTIPB  
DSNZPxxx: DSN6FAC RLFERRD

Specify what action DB2 takes if the governor encounters a condition that prevents it from accessing the resource limit specification table or if it cannot find a row in

the table that applies to the authorization ID, the plan or package name, and the logical unit of work name of the query user.

- NOLIMIT allows all dynamic SQL statements to run without limit.
- NORUN terminates all dynamic SQL statements immediately with a SQL error code.
- A number from 1 to 5000000 is the default limit; if the limit is exceeded, the SQL statement is terminated. For guidelines in choosing the default limit, see Section 5 (Volume 2) of *Administration Guide*.

For more information on using the governor for remote queries, see Section 5 (Volume 2) of *Administration Guide*.

## 6. RESYNC INTERVAL

Acceptable values: 1-99  
Default: 2  
Update: option 26 on panel DSNTIPB  
DSNZPxxx: DSN6FAC RESYNC

#

Specify the time interval, in minutes, between resynchronization periods. A resynchronization period is the time during which indoubt logical units of work involving this DB2 subsystem and partner logical units are processed.

## 7. DDF THREADS

Acceptable values: ACTIVE, INACTIVE  
Default: ACTIVE  
Update: option 26 on panel DSNTIPB  
DSNZPxxx: DSN6FAC CMTSTAT

#

Specify whether to make a thread active or inactive after it successfully commits or rolls back and holds no database locks or cursors. See Section 5 (Volume 2) of *Administration Guide* for more information about active and inactive threads.

#  
#  
#

If you specify ACTIVE, then the thread remains active. This provides the best performance but consumes system resources. If your installation must support a large number of connections, specify INACTIVE.

#  
#  
#  
#

In order for a thread to become INACTIVE, the following criteria must be met:

- no held cursors
- no held cursors have ever been left open during a commit
- package bound RELEASE(COMMIT)

## 8. DB2 GENERIC LUNAME

Acceptable values: 1-8 alphanumeric characters  
Default: none  
DSNZPxxx: none

#  
#

Specify a generic LUNAME to identify this DB2 subsystem or data sharing group in a network. You can only use a generic LU name if DB2 is running as part of an MVS Sysplex. Using a generic LUNAME helps you control the distributed workload among the servers in a data sharing group. Previously, you could associate only

one LUNAME with a LOCATION name. Now, you can associate multiple NETID.LUNAME values with a single LOCATION name. When an application requests access to a particular LOCATION, DB2 uses the SYSIBM.LOCATIONS and SYSIBM.LULIST tables to find the available network destinations (LUNAMES) for that LOCATION. See “Chapter 3-1. Connecting Distributed Database Systems” on page 3-3 for more information about setting up a generic LU name.

## 9. IDLE THREAD TIMEOUT

Acceptable values: 0-9999  
Default: 0  
Update: option 26 on panel DSNTIPB  
DSNZPxxx: DSN6FAC IDHTOIN

#

Specify the approximate time, in seconds, that an active server thread is allowed to remain idle before it is canceled. The thread is canceled after the timeout value expires, releasing its locks and cursors. Inactive and indoubt threads are not subject to timeout. Because threads are checked every 3 minutes to see if they have exceeded the timeout value, timeout values of less than 3 minutes may not be honored for up to 3 minutes.

Specifying 0 disables timeout processing. If you accept the default value, 0, DB2 continues to operate as it did prior to DB2 Version 4. (Idle server threads remain in the system and continue to hold their resources, if any.)

## 10. EXTENDED SECURITY

Acceptable values: YES, NO  
Default: NO  
Update: option 26 on panel DSNTIPB  
DSNZPxxx: DSN6SYSP EXTSEC

#

|

This field controls two related security options.

#

If you specify YES:

#

#

#

#

#

- Detailed reason codes are returned to a DRDA level 3 client when a DDF connection request fails because of security errors. When using SNA protocols, the requester must have included a product that supports the extended security sense codes. One such product is DB2 Connect, Version 5 and subsequent releases.
- RACF users can change their passwords using the DRDA change password function. This support is only for DRDA level 3 requesters that have implemented support for changing passwords.

#

#

#

#

We strongly recommend that you specify a value of **YES**. This allows properly enabled DRDA clients to determine the cause of security failures without requiring DB2 operator support. It also allows RACF users on properly enabled DB2 clients to change their passwords.

#

#

Specifying NO returns generic error codes to the clients and prevents RACF users from changing their passwords.

## Distributed Data Facility Panel: DSNTIP5

```
DSNTIP5 DISTRIBUTED DATA FACILITY PANEL 2
====> _

Enter data below:

 1 DRDA PORT ====> TCP/IP port number for DRDA clients.
 1-65534. 446 is reserved for DRDA.
 2 RESYNC PORT ====> TCP/IP port for 2-phase commit. 1-65534
 3 TCP/IP ALREADY VERIFIED ====> NO Accept requests containing only a
 userid (no password)? YES or NO

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 54. Distributed Data Facility Panel: DSNTIP5

### 1. DRDA PORT

Acceptable values: 1-65534  
Default: none  
Update: option 27 on panel DSNTIPB  
DSNZPxxx: none

Specify the TCP/IP port number used for accepting TCP/IP connection requests from remote DRDA clients. If you are enabling data sharing, each member must have the same DRDA port number. You must specify a value to use TCP/IP. Leaving this field blank means that you are not using TCP/IP. A blank field is equivalent to using 0 in the Change Log Inventory (DSNJU003) utility. See Section 3 of *Utility Guide and Reference* for more information about the Change Log Inventory utility.

### 2. RESYNC PORT

Acceptable values: 1-65534  
Default: none  
Update: option 27 on panel DSNTIPB  
DSNZPxxx: none

Specify the TCP/IP port number used to process requests for 2-phase commit resynchronization. This value must be different than the value specified for DRDA PORT. If you are enabling data sharing, each member must have a unique resynchronization port. Leaving this field blank means that you are not using TCP/IP. A blank field is equivalent to using 0 in the Change Log Inventory (DSNJU003) utility. See Section 3 of *Utility Guide and Reference* for more information about the Change Log Inventory utility.

### 3. TCP/IP ALREADY VERIFIED

|  
|  
#  
|

Acceptable values: YES, NO  
Default: NO  
Update: option 27 on panel DSNTIPB  
DSNZPxxx: DSN6FAC TCPALVER

|  
|  
|  
|  
|  
|

Specify whether TCP/IP connection requests containing only a user ID (no password, RACF PassTicket, or DCE ticket) are accepted by DB2. YES means a connection request is accepted with a user ID only. This value must be the same for all members of a data sharing group. This option applies to all incoming requests using TCP/IP regardless of the requesting location. See Section 3 (Volume 1) of *Administration Guide* for more information.

## Stored Procedures Parameters Panel: DSNTIPX

The entries on this panel are used to generate a JCL procedure for starting the stored procedures address space. See “Enabling stored procedures after installation” on page 2-263 if you want to enable stored procedures outside of the standard installation steps.

```
DSNTIPX INSTALL DB2 - STORED PROCEDURES PARAMETERS
===> _

Scrolling backward may change fields marked with asterisks
Enter data below:

* 1 WLM PROC NAME ===> ssnWLM WLM-established stored procedure
 JCL PROC name
* 2 DB2 PROC NAME ===> ssnSPAS DB2-established stored procedure
 JCL PROC name
 3 NUMBER OF TCBS ===> 8 Number of concurrent TCBS (1-100)
 4 MAX ABEND COUNT ===> 0 Allowable ABENDs for a procedure (0-255)
 5 TIMEOUT VALUE ===> 180 Seconds to wait before SQL CALL fails
 5-1800 or NOLIMIT (no timeout occurs)

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 55. Stored Procedures Parameters Panel: DSNTIPX

### 1. WLM PROC NAME

Acceptable values: 1-8 alphanumeric characters  
Default: ssnWLM  
DSNZPxxx: none

Specify a name for the stored procedures JCL procedure that is generated during installation. This procedure is used for a WLM-established stored procedures address space. If you blank out the name, the JCL procedure is still generated.

### 2. DB2 PROC NAME

Acceptable values: 1-8 alphanumeric characters  
Default: ssnSPAS  
DSNZPxxx: DSN6SYSP STORPROC

Specify a name for the JCL procedure that is used to start the DB2-established address space. If you replace the default value with blanks, you cannot start the DB2-established stored procedures address space until you update the subsystem parameter. In addition, you cannot edit the stored procedures sample jobs, DSNTSJ6S, DSNTSJ6P, DSNTSJ6T, DSNTSJ6D.

### 3. NUMBER OF TCBS

Acceptable values: 1-100  
Default: 8  
DSNZPxxx: none

Specify how many SQL CALL statements can be processed concurrently in one address space. The larger the value, the more stored procedures you can run concurrently in one address space. For information about how this value affects storage below the 16MB line, see Section 5 (Volume 2) of in *Administration Guide*

### 4. MAX ABEND COUNT

Acceptable values: 0-225  
Default: 0  
Update: option 28 on panel DSNTIPB  
DSNZPxxx: DSN6SYSP STORMXAB

Specify the number of times a stored procedure is allowed to terminate abnormally, after which SQL CALL statements for the stored procedure are rejected. The default, 0, means that the first abend of a stored procedure causes SQL CALLs to that procedure to be rejected. For production systems, you should accept the default.

### 5. TIMEOUT VALUE

Acceptable values: 5-1800  
Default: 180  
Update: option 28 on panel DSNTIPB  
DSNZPxxx: DSN6SYSP STORTIME

Specify the number of seconds before DB2 ceases to wait for an SQL CALL to be assigned to one of the task control blocks (TCBs) in a DB2 stored procedures address space. If the time interval expires, the SQL statement fails. The default is a reasonable waiting time for most sites. You might want to choose a higher value if your system has long queues. You might want to choose a lower value if you want to minimize the waiting time for end-user requests. The NOLIMIT value means that DB2 waits indefinitely for the SQL request to complete, while the thread is active. Generally, you do not want to select the NOLIMIT value because if the stored procedure address space is down for some reason, your SQL request is hung until the request is satisfied or the thread is cancelled.



## Data Definition Control Support Panel: DSNTIPZ

The entries on this panel allow you to install and tailor data definition control support. Two SQL tables (application registration and object registration) are identified and created even if data definition control support is not installed. This simplifies future activation of the facility. Specified application identifiers (DB2 plans or collections of packages) can be registered in the application registration table and, optionally, their associated DB2 object names registered in the object registration table. DB2 consults these two tables prior to accepting a given DDL statement to make sure that a particular application identifier and object name are registered. For guidance on data definition control support, see Section 3 (Volume 1) of *Administration Guide*.

```
DSNTIPZ INSTALL DB2 - DATA DEFINITION CONTROL SUPPORT
====>

Enter data below:

 1 INSTALL DD CONTROL SUPT. ====> NO YES - activate the support
 NO - omit DD control support
 2 CONTROL ALL APPLICATIONS ====> NO YES or NO
 3 REQUIRE FULL NAMES ====> YES YES or NO
 4 UNREGISTERED DDL DEFAULT ====> ACCEPT Action for unregistered DDL:
 ACCEPT - allow it
 REJECT - prohibit it
 APPL - consult ART
 Used in ART/ORT Searches
 5 ART/ORT ESCAPE CHARACTER ====>
 6 REGISTRATION OWNER ====> DSNRGCOL Qualifier for ART and ORT
 7 REGISTRATION DATABASE ====> DSNRGFDB Database name
 8 APPL REGISTRATION TABLE ====> DSN_REGISTER_APPL Table name
 9 OBJT REGISTRATION TABLE ====> DSN_REGISTER_OBJT Table name

Note: ART = Application Registration Table
 ORT = Object Registration Table

PRESS: ENTER to continue RETURN to exit HELP for more information
```

Figure 56. Data Definition Control Support Panel: DSNTIPZ

### 1. INSTALL DD CONTROL SUPT.

Acceptable values: YES, NO  
Default: NO  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFINSTL

#

Specify whether or not to install data definition control support. If NO is specified, DDL statements are not validated by this support. The application registration table and object registration table are still created according to values entered in fields 5 through 8.

### 2. CONTROL ALL APPLICATIONS

Acceptable values: YES, NO  
Default: NO  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFDEDPL

#

Specify whether or not the DB2 subsystem is completely controlled by a set of closed applications whose application identifiers are identified in the application registration table. Closed applications require their DB2 objects to be managed solely through the plans or packages of the closed application registered in the application registration table.

### 3. REQUIRE FULL NAMES

# Acceptable values: YES, NO  
Default: YES  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFFULLQ

Specify whether or not registered objects require fully qualified names.

### 4. UNREGISTERED DDL DEFAULT

# Acceptable values: ACCEPT, REJECT, APPL  
Default: ACCEPT  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFDEFLT

Specify what action is taken for DDL that names an unregistered object. If the ACCEPT option is specified, the DDL is accepted. If the REJECT option is specified, the DDL is rejected. If APPL is specified, the DDL is rejected if the current application is not registered.

### 5. ART/ORT ESCAPE CHARACTER

# Acceptable values: any non-alphanumeric character  
Default: none  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFESCP

Specify the escape character to be used in the application registration table (ART) or object registration table (ORT). Sets of names in the ART and ORT can be represented by patterns that use the underscore(\_) and percent sign (%) characters in the same way as in an SQL LIKE predicate.

If you enter a character in this field, it can be used in those patterns in the same way as an escape character is used in an SQL LIKE predicate. See Section 3 (Volume 1) of *Administration Guide* for examples of using the percent and underscore characters and the escape character.

### 6. REGISTRATION OWNER

# Acceptable values: 1-8 characters  
Default: DSNRGCOL  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFCOLID

Specify the owner of the application registration table and the object registration table.

## 7. REGISTRATION DATABASE

# Acceptable values: 1-8 characters  
Default: DSNRGFDB  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFDBNAM

Specify the name of the database which contains the registration tables.

## 8. APPL REGISTRATION TABLE

# Acceptable values: 1-17 characters  
Default: DSN\_REGISTER\_APPL  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFNMPRT

Specify the name of the application registration table.

## 9. OBJT REGISTRATION TABLE

# Acceptable values: 1-17 characters  
Default: DSN\_REGISTER\_OBJT  
Update: option 29 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RGFNMORT

Specify the name of the object registration table.

---

## Job Editing Panel: DSNTIPY

The entries on this panel specify values and information about job statements for the installation and sample application jobs.

**Establishing System Affinity for Installation Jobs:** You must ensure that the installation jobs run on the MVS system where the appropriate DB2 subsystem is running. There are several MVS installation-specific ways to make sure this happens. These include:

- For JES2 multi-access spool (MAS) systems, use the following JCL statement:

```
/*JOBPARM SYSAFF=cccc
```

Where *cccc* is the JES2 name. You can specify an asterisk (SYSAFF=\*) to indicate that the job should run on the system from which it was submitted.

- For JES3 systems, use the following JCL statement:

```
//*MAIN SYSTEM=(main-name)
```

Where *main-name* is the JES3 name.

*OS/390 MVS JCL Reference* describes the above JCL statements. You can edit the jobs manually, or you can enter the above statements on installation panel DSNTIPY and have DB2 insert these statements for you.

Your installation might have other mechanisms for controlling where batch jobs run, such as by using job classes.

**Ensuring that Installation Jobs Access the Right JCL Procedures:** If your MVS system has more than one procedure library, you need to ensure that your installation jobs access the correct set of procedures. One way is to use a JCLLIB statement to specify the order for procedure libraries.

The JCLLIB statement has the following form:

```
//ddname JCLLIB ORDER=(library[,library...])
```

The JCLLIB statement must follow the JOB statement and precede the first EXEC statement in the job. If you enter this statement on panel DSNTIPY, DB2 will insert it into your JCL.

For more information on the JCLLIB statement, see *OS/390 MVS JCL Reference*.

```

DSNTIPY INSTALL DB2 - JOB EDITING
====>

Enter data below:

 1 REMOTE LOCATION ====> Remote location for COBOL
 organization application
 2 COBOL TYPE ====> IBMCOB COBOL for sample applications
 (COBOL, COB2, or IBMCOB)

Enter job card information for install and sample jobs:

 3 ====>
 4 ====>
 5 ====>
 6 ====>
 7 ====>
 8 ====>

PRESS: ENTER to continue RETURN to exit HELP for more information

```

Figure 57. Job Editing Panel: DSNTIPY

## 1. REMOTE LOCATION

Acceptable values: 1-16 characters  
 Default: none  
 Update: See "The Update Process" on page 2-228  
 DSNZPxxx: none

Specify the name of another location to be used by the COBOL preparation sample job (DSNTEJ3C), the DDF remote location update sample job (DSNTEJ6), and the stored procedures sample jobs (DSNTEJ6S, DSNTEJ6P, DSNTEJ6T and DSNTEJ6D). The name must begin with a letter and must not contain special characters. A remote location name is accepted only if you have also entered a DB2 location name for DB2 LOCATION NAME (field 2 on installation panel DSNTIPR).

## 2. COBOL TYPE

Acceptable values: COBOL, COB2, IBMCOB  
 Default: IBMCOB  
 Update: see Update on page 2-228  
 DSNZPxxx: none

Specify which COBOL release to include when editing COBOL sample application jobs (DSNTEJ2C, DSNTEJ3C, DSNTEJ4C, and DSNTEJ5C). See "Special Considerations for COBOL Programs" on page 2-308 for notes on how to convert the tailored sample jobs if you want to test more than one type of COBOL.

### 3-8 job card information

|                    |                          |
|--------------------|--------------------------|
| Acceptable values: | see below                |
| Default:           | none                     |
| Update:            | see Update on page 2-228 |
| DSNZPxxx:          | none                     |

Specify the job statements used in all the installation and sample application jobs.

The job name can be specified in one of two ways:

- If the job name is *member*, the job name for each job is the same as its member name.
- If the job name is any value other than *member*, the name is truncated to seven characters and one character is added to the end of the name identifying the run order for that job.

---

## Install DB2—CLIST Calculations Panel 1: DSNTIPC

This panel displays the messages produced by the installation CLIST indicating calculated storage sizes. Space estimates from these messages do not account for cylinder rounding. Base requirements can be 10 to 20% higher than the message indicates depending on the DASD type. If you need more information about these messages, see Section 3 of *Messages and Codes*.

The messages show that most of the needed virtual storage is in extended private storage (including the buffer pool, the EDM pool, most of the code, and a significant amount of working storage).

During the tailoring session, a warning message is issued to the tailoring terminal. This message is always issued if you accept the default.

```
DSNT438I WARNING, IRLM LOCK MAXIMUM SPACE = ir1mreg K, AVAILABLE = ir1mav K
```

This message shows that the IRLM could request a total amount of space larger than the available space, causing an abend. The message is based on the maximum number of page or row locks per user specified on installation panel DSNTIPJ (LOCKS PER USER) and the number of users specified on installation panel DSNTIPE for MAX USERS and MAX REMOTE ACTIVE during the tailoring session. The formula is:

$$(\text{MAX USERS} + \text{MAX REMOTE ACTIVE}) \times \text{LOCKS PER USER} \times 250 \text{ bytes per lock}$$

The CLIST assumes that the private region available for IRLM locks is estimated as 60000KB, if extended private address space is used or the amount of space specified on parameter MAXIMUM ECSA is the default of 6MB.

When using the default in the tailoring session you get:

$$70 \times 10000 \times 250 = 175000\text{KB}$$

This amount is a high-end estimate. It is the amount of storage needed if the maximum number of users are connected and each uses the maximum number of locks. Most users hold only a few locks.





### 3. BUFFER POOL SIZE

Acceptable values: none  
Default: based on calculations  
Update: run CLIST again  
DSNZPxxx: none

This field specifies the buffer pool size calculated by the CLIST. Field 2 is protected and cannot be changed during update mode. If you want to change the size of a buffer pool, you must use the -ALTER BUFFERPOOL command as described in Chapter 2 of *Command Reference*.

### 4. SORT POOL SIZE

Acceptable values: 240K-64000K  
Default: based on calculations  
Update: press ENTER twice on panel DSNTIPB  
DSNZPxxx: DSN6SPRM SRTPOOL

This field specifies the amount of storage needed for the sort pool as calculated by the CLIST. You have a choice of:

- Accepting the value in the Calculated column; the CLIST calculates this value based on input from previous panels. If there is a value in the Override column, you must erase the override value in order to accept the calculated value.
- Typing your own value in the Override column.

If you decide to change this field, estimate the sort pool value with the following formula:

#  $16000 \times (12 + \text{sort key length} + \text{sort data length})$   
#  $+ 4$  (if ESA hardware sort assist))

See Section 5 (Volume 2) of *Administration Guide* for detailed instructions on choosing sizes for optimal performance.

### 5. RID POOL SIZE

Acceptable values: 0, 16K-1000000K  
Default: based on calculations  
Update: press ENTER twice on panel DSNTIPB  
DSNZPxxx: DSN6SPRM MAXRBLK

This field specifies the amount of storage needed for the RID pool as calculated by the CLIST. You have a choice of:

- Accepting the value in the Calculated column; the CLIST calculates this value based on input from previous panels. If there is a value in the Override column, you must erase the override value in order to accept the calculated value.
- Type in your own value in the Override column.

If you decide to change this field, estimate the storage required for the RID pool with the following formula:

Number of concurrent RID processing activities  $\times$   
average number of RIDs  $\times 2 \times 4$  (bytes per RID)

Choosing '0' disables the use of the RID pool. In this case, DB2 does not use access paths or join methods that depend on RID pool storage.

See Section 5 (Volume 2) of *Administration Guide* for how to choose a RID pool size for optimal performance.

#### 6-10. **CLIST messages**

|                    |                 |
|--------------------|-----------------|
| Acceptable values: | none            |
| Default:           | none            |
| Update:            | run CLIST again |
| DSNZPxxx:          | none            |

These fields specify sizes calculated by the CLIST. Fields 6 through 10 are protected and cannot be changed.

#### 11-12. **Storage messages**

|                    |                 |
|--------------------|-----------------|
| Acceptable values: | none            |
| Default:           | none            |
| Update:            | run CLIST again |
| DSNZPxxx:          | none            |

These fields are the results of the calculations described on the previous page. These fields are protected and cannot be changed.

---

## Install DB2—CLIST Calculations Panel 2: DSNTIPC1

```
DSNTIPC1 INSTALL DB2 - CLIST CALCULATIONS - PANEL 2
===> _

1 DSNT488I VOLUME DSNV01 WILL REQUIRE AT LEAST 47878 4K BLOCKS
2 DSNT488I VOLUME DSNV02 WILL REQUIRE AT LEAST 41163 4K BLOCKS
3 DSNT488I VOLUME SYS103 WILL REQUIRE AT LEAST 58278 4K BLOCKS
4 DSNT488I VOLUME SYS104 WILL REQUIRE AT LEAST 4308 4K BLOCKS
5 DSNT488I VOLUME SYS105 WILL REQUIRE AT LEAST 25995 4K BLOCKS
6 DSNT488I VOLUME SYS106 WILL REQUIRE AT LEAST 25995 4K BLOCKS

PRESS: ENTER to select RETURN to exit HELP for more information
```

Figure 59. CLIST Calculations Panel 2: DSNTIPC1

### 1-6. CLIST messages

|                    |                 |
|--------------------|-----------------|
| Acceptable values: | none            |
| Default:           | none            |
| Update:            | run CLIST again |
| DSNZPxxx:          | none            |

These CLIST messages may or may not appear depending on how many unique volume names you supplied on installation panel DSNTIPA2.

---

## Completing the CLIST Processing

After receiving the CLIST messages on installation panel DSNTIPC1 indicating sizes calculated, press ENTER to begin CLIST processing. You then receive a series of messages that detail the CLIST processing. If you need more information about these messages, see Section 3 of *Messages and Codes*.

## Responding to Messages

You first receive the following message:

```
DSNT478I BEGINNING EDITED DATA SET OUTPUT
```

The CLIST is checking the parameter values you entered. If it detects a problem, you receive an error or warning message indicating the name of the parameter and the type of problem. If you receive an *error* message, the CLIST cannot edit the installation or migration jobs properly. If you receive a warning message, check the conditions. It is possible to receive a warning message when the conditions are normal or acceptable. If you specify several large numbers in the panels, the CLIST might send a message indicating an overflow in CLIST arithmetic.

At this point the CLIST displays the Main Panel again. You can proceed through the panels, rechecking or changing parameter values.

If the CLIST does not find any errors, you receive messages indicating the amount of disk storage and virtual storage that is needed. For information about installation panel DSNTIPC1, which displays these messages, see “Install DB2—CLIST Calculations Panel 1: DSNTIPC” on page 2-215. (You might also receive some other information messages.)

## Tailoring the Installation Jobs

The CLIST tailors each job according to the panel values you specified. For each edited job, you receive the following message:

```
DSNT489I CLIST EDITING dsname(member), explanation
```

### Attention:

If an error occurs while the installation CLIST edits your jobs, you will receive this message:

```
IKJ52555I NOTHING SAVED
ENTER SAVE OR END-
```

Enter END to prevent modification of the original copies of your installation jobs.

After the CLIST finishes tailoring the jobs, it displays the Main Panel again. If you need to continue your tailoring at another time, conclude this session. Then, when you start a new session, use the value you specified for OUTPUT MEMBER NAME during this session as the value for INPUT MEMBER NAME during the new session. Enter these values on the Main Panel.

If you receive a message from the editor, such as TEXT NOT FOUND, enter END NOSAVE to exit. That message can indicate an error. You can rerun the CLIST with the trace control parameter set to CONTROL(SYMLIST) to learn what caused the problem. In some cases, specifying CONTROL(LIST) as the trace control parameter may provide enough information for you to find the source of the problem.

The installation CLIST uses the values you specify on the installation panels to tailor and load the installation or migration jobs. Each job is composed of one or more JCL procedures or job steps. The CLIST loads each job as a separate member of the newly created *prefix.NEW.SDSNSAMP*. Before you run any of these jobs, however, you might want to perform some editing that is not done by the CLIST. If DASD allocation is completely controlled by SMS for your installation, verify that the input to IDCAMS from the install jobs does not conflict with the requirements for SMS.

This section identifies several items you may want to add or change in the jobs. These changes are general; that is, they apply to all the jobs processed by the CLIST. Later chapters explain changes you can make for specific jobs.

Which jobs you edit depends on the task you are performing: installation, migration, or update. In data sharing environments, different jobs are edited depending on the data sharing function: group, member, or enable. The installation CLIST tailors a different set of jobs for each task.

**If you are installing**, the CLIST tailors these jobs:

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| DSNTIJMV | DSNTIJCA | DSNTIJIN | DSNTIJID | DSNTIJUZ |
| DSNTIJEX | DSNTIJVC | DSNTIJSU | DSNTIJTM | DSNTIJSG |
| DSNTIJIC | DSNTIJDE | DSNTEJ0  | DSNTEJ1  | DSNTEJ1P |
| DSNTEJ1S | DSNTEJ1T | DSNTEJ2A | DSNTEJ2C | DSNTEJ2D |
| DSNTEJ2E | DSNTEJ2F | DSNTEJ2P | DSNTEJ3C | DSNTEJ3P |
| DSNTEJ4C | DSNTEJ4P | DSNTEJ4P | DSNTEJ3C | DSNTEJ3P |
| DSNTESE  |          | DSNTEJ4P | DSNTEJ3C | DSNTEJ3P |

If you have activated DDF, the CLIST also tailors job DSNTEJ6.

If you have activated stored procedures, DSNTEJ6D, DSNTEJ6S, DSNTEJ6P, and DSNTEJ6T are edited.

If CICSREL is 0, the CLIST edits DSNTEJ5A, DSNTEJ5C, DSNTEJ5P, and DSN8FRDO.

If you are using data sharing, DSNTIJGF and DSNTIJFT are also edited.

The installation CLIST tailors the DSNH, DSNU, and DSNEMC01 CLISTs for installation.

**If you are migrating**, the CLIST tailors the following jobs:

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| DSNTEJ0  | DSNTEJ1  | DSNTEJ1P | DSNTEJ1S | DSNTEJ1T |
| DSNTEJ2A | DSNTEJ2C | DSNTEJ2D | DSNTEJ2E | DSNTEJ2F |
| DSNTEJ2P | DSNTEJ3C | DSNTEJ3P | DSNTEJ4C | DSNTEJ4P |
| DSNTEJ4C | DSNTEJ4P | DSNTEJ4P | DSNTEJ4C | DSNTEJ4P |
| DSNTEJ4P | DSNTEJ4P | DSNTEJ4P | DSNTEJ4C | DSNTEJ4P |
| DSNTEJ4P | DSNTEJ4P | DSNTEJ4P | DSNTEJ4C | DSNTEJ4P |
| DSNTEJ4P | DSNTEJ4P | DSNTEJ4P | DSNTEJ4C | DSNTEJ4P |
| DSNTEJ4P | DSNTEJ4P | DSNTEJ4P | DSNTEJ4C | DSNTEJ4P |
| DSNTEJ4P | DSNTEJ4P | DSNTEJ4P | DSNTEJ4C | DSNTEJ4P |
| DSNTEJ4P | DSNTEJ4P | DSNTEJ4P | DSNTEJ4C | DSNTEJ4P |

If you have activated DDF, the CLIST also tailors job DSNTEJ6. These tailored CLISTs are found in *prefix.NEW.SDSNTEMP*.

If you have activated stored procedures, DSNTEJ6D, DSNTEJ6S, DSNTEJ6P, and DSNTEJ6T are edited.

If CICSREL is not 0, DSN8FRDO, DSNTEJ5A, DSNTEJ5C, and DSNTEJ5P are edited.

The CLIST also tailors the DSNH, DSNHC, DSNU, and DSNEMC01 CLISTs for migration.

**If you are updating**, the CLIST tailors only one job: DSNTIJUZ.

These jobs are described in the following chapters. Recovery information is provided, along with a description of each job. Unless otherwise stated in the job description, a return code of 0 or 4 from any of the jobs indicates successful completion. Some of the jobs contain statements that could fail without causing the job to fail. For instance, delete commands for data sets, drop statements for SQL objects, and stop commands could fail when you first run a job because the data sets or objects do not exist. Unless otherwise stated, you can ignore these failures. The statements are needed to allow you to rerun the job (if necessary) without performing the deletes, drops, and stops manually; they are merely for cleanup or initialization processing.

When a job fails, follow the instructions provided in the recovery information for the job. If you need further recovery information, refer to *Messages and Codes* and examine the descriptions of the messages that the job generated.

Before you begin editing, you might want to print or back up the jobs. You can print the JCL for these jobs using IEBTPCH or any other print facility available at your site.

Consider the following suggestions for possible changes or additions to the jobs:

1. Tailor the jobs to the needs of your site. You should edit the jobs to conform to any unique requirements you might have. Also, you might want to make any minor JCL changes for items that were not handled by the ISPF panels.
2. Examine the volume serial numbers used in the various jobs. The volume serial number fields of installation panel DSNTIPA2 allow you to specify up to seven volumes for the data sets defined during installation or migration. If you want to use more than seven volumes, specify them now.

The DSNTINST CLIST spreads the data sets across the volumes you specify. Adding more volumes to provide more separation of data sets can help improve system performance and recoverability. Many of the log data sets are large and easy to place on separate volumes. The CLIST produces a series of messages that estimate space distribution for the volumes specified. For more information, see “Install DB2—CLIST Calculations Panel 1: DSNTIPC” on page 2-215.

3. Edit the DSNH CLIST if needed. The DSNH CLIST allows you to precompile, compile, prelink-edit, link-edit, bind, and run an application by issuing a single command. You might need to edit the DSNH CLIST to change values for some of the entries. Ensure that all DSNH keyword parameters for all DB2-supported compilers are checked and are correct for your applications. For a description of the parameters, see the DSNH CLIST in Chapter 2 of *Command Reference*.
  - Check the default data set names for the licensed programs you have installed. These defaults are in the parameter definitions at the beginning of each program. If the names and prefixes are not correct for your site, change them.
  - Check default library names. If the names and prefixes are not correct for your site, change them. Make sure the data sets exist and are cataloged for BLIB, CLIB, LLIB, and PLIB. When the DSNH CLIST runs, it creates DBRMLIB and LOAD data sets if they do not already exist. The DBRMLIB data set is only created if the DBRMLIB(DEFAULT) is set. The following are the default library names:

BLIB(NONE)

DBRMLIB(DEFAULT)—Because the DBRM library must be allocated exclusively when the precompiler writes to it, we recommend having a temporary library, or one per user, rather than trying to share libraries. However, if your DB2 subsystem uses DFSMSdftp's partitioned data set extended (PDSE) for managing data sets, access is restricted at member level rather than data set level; this provides another alternative for concurrent access to the DBRM library.

CLIB(NONE)

LLIB(NONE)

LOAD(RUNLIB.LOAD)—Because this library is allocated exclusively when it is being written, we recommend having a temporary library, or one per user, rather than trying to share libraries.



Table 58 (Page 2 of 4). Directory of Subsystem Parameters

| DSN6xxx Macro<br>Parameter | Macro    | Panel    | Item<br>Number | Page    |
|----------------------------|----------|----------|----------------|---------|
| CACHEPAC                   | DSN6SPRM | DSNTIPP  | 16             | 2-181   |
| CATALOG                    | DSN6ARVP | DSNTIPA  | 4              | 2-193   |
| "                          | DSN6SPRM | DSNTIPA2 | 1              | 2-105   |
| CHGDC                      | DSN6SPRM | DSNTIPO  | 10             | 2-158   |
| CMTSTAT                    | DSN6FAC  | DSNTIPR  | 7              | 2-201   |
| COMPACT                    | DSN6ARVP | DSNTIPA  | 15             | 2-193   |
| CONDBAT                    | DSN6SYSP | DSNTIPE  | 4              | 2-145   |
| COORDNTR                   | DSN6GRP  | DSNTIPK  | 5              | 2-110   |
| CDSSRDEF                   | DSN6SPRM | —        | —              | Note 2  |
| CTHREAD                    | DSN6SYSP | DSNTIPE  | 2              | 2-145   |
| DDF                        | DSN6FAC  | DSNTIPR  | 1              | 2-201   |
| DEALLCT                    | DSN6LOGP | DSNTIPA  | 9              | 2-193   |
| DECDIV3                    | DSN6SPRM | DSNTIPF  | 3              | 2-164   |
| DEFIXTP                    | DSN6SPRM | DSNTIPE  | 9              | 2-145   |
| DEFLTID                    | DSN6SPRM | DSNTIPP  | 12             | 2-181   |
| DESCSTAT                   | DSN6SPRM | —        | —              | See 3-4 |
| DLDFREQ                    | DSN6SYSP | —        | —              | Note 3  |
| DLITOUT                    | DSN6SPRM | —        | —              | Note 4  |
| DSHARE                     | DSN6GRP  | DSNTIPA1 | 2              | 2-99    |
| DSMAX                      | DSN6SPRM | DSNTIPC  | 1              | 2-45    |
| EDMPOOL                    | DSN6SPRM | DSNTIPC  | 2              | 2-215   |
| EDPROP                     | DSN6SPRM | DSNTIPO  | 10             | 2-158   |
| EXTSEC                     | DSN6SYSP | DSNTIPR  | 10             | 2-201   |
| GRPNAME                    | DSN6GRP  | DSNTIPK  | 1              | 2-110   |
| HOPAUTH                    | DSN6SPRM | —        | —              | Note 5  |
| IDBACK                     | DSN6SYSP | DSNTIPE  | 6              | 2-145   |
| IDFORE                     | DSN6SYSP | DSNTIPE  | 5              | 2-145   |
| IDTHTOIN                   | DSN6FAC  | DSNTIPR  | 9              | 2-201   |
| INBUFF                     | DSN6LOGP | DSNTIPL  | 2              | 2-190   |
| IRLMAUT                    | DSN6SPRM | DSNTIPI  | 5              | 2-174   |
| IRLMPRC                    | DSN6SPRM | DSNTIPI  | 6              | 2-174   |
| IRLMRWT                    | DSN6SPRM | DSNTIPI  | 3              | 2-174   |
| IRLMSID                    | DSN6SPRM | DSNTIPI  | 2              | 2-174   |
| IRLMSWT                    | DSN6SPRM | DSNTIPI  | 7              | 2-174   |
| LOGLOAD                    | DSN6SYSP | DSNTIPN  | 9              | 2-154   |
| MAXARCH                    | DSN6LOGP | DSNTIPA  | 10             | 2-193   |
| MAXDBAT                    | DSN6SYSP | DSNTIPE  | 3              | 2-145   |
| MAXKEEPD                   | DSN6SPRM | DSNTIPE  | 10             | 2-145   |
| MAXRBLK                    | DSN6SPRM | DSNTIPC  | 4              | 2-215   |
| MAXRTU                     | DSN6LOGP | DSNTIPA  | 8              | 2-193   |
| MEMBNAME                   | DSN6GRP  | DSNTIPK  | 2              | 2-110   |
| MON                        | DSN6SYSP | DSNTIPN  | 7              | 2-154   |
| MONSIZE                    | DSN6SYSP | DSNTIPN  | 8              | 2-154   |
| NUMLKTS                    | DSN6SPRM | DSNTIPJ  | 3              | 2-177   |
| NUMLKUS                    | DSN6SPRM | DSNTIPJ  | 4              | 2-177   |
| OJPERFEH                   | DSN6SPRM | —        | —              | Note 6  |
| OUTBUFF                    | DSN6LOGP | DSNTIPL  | 3              | 2-190   |
| PCLOSEN                    | DSN6SYSP | —        | —              | Note 7  |
| PCLOSET                    | DSN6SYSP | —        | —              | Note 8  |
| PRIQTY                     | DSN6ARVP | DSNTIPA  | 2              | 2-193   |
| PROTECT                    | DSN6ARVP | DSNTIPP  | 5              | 2-181   |
| QUIESCE                    | DSN6ARVP | DSNTIPA  | 14             | 2-193   |
| RECALL                     | DSN6SPRM | DSNTIPO  | 2              | 2-158   |
| RECALLD                    | DSN6SPRM | DSNTIPO  | 3              | 2-158   |
| RELCURHL                   | DSN6SPRM | —        | —              | Note 9  |
| RESTART/DEFER              | DSN6SPRM | DSNTIPS  | 1              | 2-199   |
| RESYNC                     | DSN6FAC  | DSNTIPR  | 6              | 2-201   |
| RETLWAIT                   | DSN6SPRM | —        | —              | Note 10 |
| RGFCOLID                   | DSN6SPRM | DSNTIPZ  | 6              | 2-209   |
| RGFDBNAM                   | DSN6SPRM | DSNTIPZ  | 7              | 2-209   |



Table 58 (Page 3 of 4). Directory of Subsystem Parameters

| DSN6xxx Macro<br>Parameter | Macro    | Panel   | Item<br>Number | Page  |
|----------------------------|----------|---------|----------------|-------|
| RGFDEDPL                   | DSN6SPRM | DSNTIPZ | 2              | 2-209 |
| RGFDEFLT                   | DSN6SPRM | DSNTIPZ | 4              | 2-209 |
| RGFESCP                    | DSN6SPRM | DSNTIPZ | 5              | 2-209 |
| RGFFULLQ                   | DSN6SPRM | DSNTIPZ | 3              | 2-209 |
| RGFINSTL                   | DSN6SPRM | DSNTIPZ | 1              | 2-209 |
| RGFNMORT                   | DSN6SPRM | DSNTIPZ | 9              | 2-209 |
| RGFNMPRT                   | DSN6SPRM | DSNTIPZ | 8              | 2-209 |
| RLF                        | DSN6SYSP | DSNTIPO | 4              | 2-158 |
| RLFAUTH                    | DSN6SYSP | DSNTIPP | 13             | 2-181 |
| RLFERR                     | DSN6SYSP | DSNTIPO | 6              | 2-158 |
| RLFERRD                    | DSN6FAC  | DSNTIPR | 5              | 2-201 |
| RLFTBL                     | DSN6SYSP | DSNTIPO | 5              | 2-158 |
| ROUTCDE                    | DSN6SYSP | DSNTIPO | 1              | 2-158 |
| RRULOCK                    | DSN6SPRM | DSNTIPI | 9              | 2-174 |
| SECQTY                     | DSN6ARVP | DSNTIPA | 3              | 2-193 |
| SEQCACH                    | DSN6SPRM | DSNTIPE | 7              | 2-145 |
| SEQPRES                    | DSN6SPRM | DSNTIPE | 8              | 2-145 |
| SITETYP                    | DSN6SPRM | DSNTIPO | 11             | 2-158 |
| SMFACCT                    | DSN6SYSP | DSNTIPN | 4              | 2-154 |
| SMFSTAT                    | DSN6SYSP | DSNTIPN | 5              | 2-154 |
| SRTPOOL                    | DSN6SPRM | DSNTIPC | 3              | 2-215 |
| STATIME                    | DSN6SYSP | DSNTIPN | 6              | 2-154 |
| STORMXAB                   | DSN6SYSP | DSNTIPX | 4              | 2-207 |
| STORPROC                   | DSN6SYSP | DSNTIPX | 2              | 2-207 |
| STORTIME                   | DSN6SYSP | DSNTIPX | 5              | 2-207 |
| SYSADM                     | DSN6SPRM | DSNTIPP | 8              | 2-181 |
| SYSADM2                    | DSN6SPRM | DSNTIPP | 9              | 2-181 |
| SYSOPR1                    | DSN6SPRM | DSNTIPP | 10             | 2-181 |
| SYSOPR2                    | DSN6SPRM | DSNTIPP | 11             | 2-181 |
| TCPALVER                   | DSN6FAC  | DSNTIP5 | 3              | 2-205 |
| TRACSTR                    | DSN6SYSP | DSNTIPN | 2              | 2-154 |
| TRACTBL                    | DSN6SYSP | DSNTIPN | 3              | 2-154 |
| TSTAMP                     | DSN6ARVP | DSNTIPH | 9              | 2-112 |
| TWOACTV                    | DSN6LOGP | DSNTIPH | 3              | 2-112 |
| TWOARCH                    | DSN6LOGP | DSNTIPH | 6              | 2-112 |
| TWOBSDS                    | DSN6LOGP | —       | —              | 2-281 |
| UNIT                       | DSN6ARVP | DSNTIPA | 5              | 2-193 |
| UNIT2                      | DSN6ARVP | DSNTIPA | 6              | 2-193 |
| URCHKTH                    | DSN6SYSP | DSNTIPN | 10             | 2-154 |
| UTIMOUT                    | DSN6SPRM | DSNTIPI | 8              | 2-174 |
| WRTHRSH                    | DSN6LOGP | DSNTIPL | 4              | 2-190 |

Table 58 (Page 4 of 4). Directory of Subsystem Parameters

| DSN6xxx Macro<br>Parameter | Macro | Panel | Item<br>Number | Page |
|----------------------------|-------|-------|----------------|------|
|----------------------------|-------|-------|----------------|------|

**Note:**

- #
- #
1. BMPTOUT specifies the IMS BMP timeout multiplier connections. See topic Installation Options for Wait Times in Section 5 (Volume 2) of *Administration Guide* for more information.
  2. CDSSRDEF is the CURRENT DEGREE special register to set query parallelism. See Section 5 (Volume 2) of *Administration Guide* for more information.
  3. DDLFREQ can be set to 0 to disable down-level detection. See Section 4 (Volume 1) of *Administration Guide* for more information.
  4. DLITOUT is the timeout multiplier for IMS DL/I connections. See topic Installation Options for Wait Times in Section 5 (Volume 2) of *Administration Guide* for more information.
  5. HOPAATH specifies which authorization scheme to use in a hop situation. See topic Privileges Exercised through a Plan or Package in Section 3 (Volume 1) of *Administration Guide* for more information.
  6. OJPERFEH activates the outer join performance enhancements. See Section 5 (Volume 2) of *Administration Guide* for additional information.
  7. PCLOSEN is the number of consecutive DB2 checkpoints since a page set or partition was last updated. See Section 5 (Volume 2) of *Administration Guide* for additional information.
  8. PCLOSET is the amount of elapsed time since a page set or partition was last updated. See Section 5 (Volume 2) of *Administration Guide* for more information.
  9. RELCURHL specifies whether an unnecessary cursor position lock is released or kept after commit on cursor defined WITH\_HOLD. The default is NO. See Section 5 (Volume 2) of *Administration Guide* for more information.
  10. RETLWAIT specifies whether to wait for an incompatible retained lock to become available. See Chapter 3 of *Data Sharing: Planning and Administration* for more information.

**Subsystem Parameters added after General Availability**

#

#

#

#

#

#

#

Some subsystem parameters are added after the product ships through APARs. These subsystem parameters are reflected in the macros but not on panels in this release. To modify these parameters, you must edit the macros in job DSNTIJUZ. After modifying the macros, job DSNTIJUZ must be assembled and link-edited. Then the DB2 subsystem must be stopped and started for the modified parameters to be used.

#

#

#

#

#

#

The new DSNHDECP parameter DYNRULS can only be specified in the DSNHDECM parameters found in step DSNTIZP of job DSNTIJUZ. The DYNRULS parameter controls whether dynamic SQL statements are parsed using the precompiler options or the application programming (DSNHDECP) options. The behavior of this parameter is discussed in *Command Reference* under DYNAMICRULES(BIND).

#

#

#

#

#

#

A field has been added to panel DSNTIPJ which controls whether IRLM automatically disconnects from the data sharing group when DB2 is not identified to it. The field called DISCONNECT IRLM does not have a subsystem parameter but can be added by editing the IRLM start procedure. The options are YES or NO, with YES being the default.

#

#

#

#

#

#

A new option, COEXIST, has been added to the AUTO BIND parameter on panel DSNTIPO. Specifying COEXIST allows automatic rebind operations to be performed in a DB2 data sharing coexistence environment only when the plan/package is marked invalid (SYSPLAN or SYSPACKAGE column VALID contains 'N') or was last bound on DB2 Version 5 and is now running on DB2 Version 4. The subsystem parameter is DSN6SPRM ABIND.

#

#

#

A new subsystem parameter, CONTRACT THREAD STG, has been added to panel DSNTIPE. This parameter specifies whether DB2 will periodically "contract" each thread's working storage area. Storage acquired by a thread is normally allocated

# to that thread until deallocation. A value of YES means storage will periodically be  
# returned to the operating system. For best performance, use a value of NO. The  
# subsystem parameter is DSN6SPRM CONSTSTOR.

# A new subsystem parameter, PTASKROL, has been added to macro DSN6SYSP.  
# This parameter indicates whether to roll up query parallel task's account trace  
# records into the originating task's accounting trace. A value of YES means the  
# originating task will cut an additional accounting trace record with all the roll up  
# values from parallel tasks. A value of NO means each parallel task will produce its  
# own accounting trace.

# A new subsystem parameter, XLKUPDLT, has been added to macro DSN6SPRM.  
# This parameter specifies the locking method used when performing a searched  
# UPDATE or DELETE. A value of NO means DB2 uses an S or U lock when  
# scanning for qualifying rows. For any qualifying rows or pages the lock is upgraded  
# to an X lock before performing the update or delete. For non-qualifying rows or  
# pages the lock is released if using ISOLATION(CS). For ISOLATION(RS) or  
# ISOLATION(RR), an S lock is retained on the rows or pages until the next commit  
# point. A value of YES means DB2 gets an X lock on qualifying rows or pages. For  
# ISOLATION(CS), the lock is released if the rows or pages are not updated or  
# deleted. For ISOLATION(RS) or ISOLATION(RR), an X lock is retained until the  
# next commit point.

# A new subsystem parameter PARTKEYU, has been added to macro DSN6SPRM.  
# This parameter specifies whether values in columns that participate in partitioning  
# keys may be updated. A value of YES means the values in such columns may be  
# updated. The default value is YES. A value of NO means the values in such  
# columns may not be updated. A value of SAME means the values in such columns  
# may be updated if and only if the update leaves the updated row in the same  
# partition as the original row.

# A new subsystem parameter RETVLCFK, has been added to macro DSN6SPRM.  
# This parameter specifies whether DB2 must go to the data page to retrieve data or  
# whether data can be retrieved from the index with padded blanks to the full  
# maximum length of the column. The default is NO. YES can mean better  
# performance, but the applications must be able to handle the padding blanks. If  
# you choose YES, you must rebind plans and packages to enable the change.

# A new subsystem parameter OJPERFEH, has been added to macro DSN6SPRM.  
# This parameter provides some enhancements for outer join performance. See  
# APAR PQ18710 for details. The default value of NO means the enhancements are  
# disabled. The value YES enables the enhancements for outer join performance.

# A new subsystem parameter PARAMDEG, has been added to macro DSN6SPRM.  
# This parameter specifies the amount of parallelism allowed for a parallel group.  
# The default value of 0 means there is no upper limit on the degree of parallelism for  
# a parallel group. A non-zero value (1-254) indicates the maximum degree of  
# parallelism for this parallel group. For performance information about enabling  
# parallel processing see Section 5 (Volume 2) of *Administration Guide*.

# A new subsystem parameter EDMBFIT, has been added to macro DSN6SPRM.  
# This parameter is used to adjust the free chain search algorithm on systems with a  
# large EDM pool (greater than 40M). The default value of NO means use a first fit  
# algorithm. The YES value means use a better fit algorithm. For performance

# information about tuning the EDM pool see Section 5 (Volume 2) of *Administration*  
# *Guide*.

---

## The Update Process

This section describes how to modify some of the parameters you specified when installing or migrating DB2. This process allows you to tailor DB2 more precisely to your needs.

The update process does not generate a complete set of installation or migration jobs, as the installation and migration process does. It generates only one job: DSNTIJUZ. This job assembles and link-edits the DB2 data-only subsystem parameter module, DSNZPARM (or the value you specified for PARAMETER MODULE on installation panel DSNTIPO), and the application program's default module, DSNHDECP.

## Updating Parameters through the Update Selection Menu Panel: DSNTIPB

To update most parameters, follow these steps:

1. Run the installation CLIST and specify UPDATE on installation panel DSNTIPA1. See Data Set Names Panel 1: DSNTIPT for information about the output data sets.
2. Choose the output SDSNSAMP data set on installation panel DSNTIPT. See page 2-116 for information about the output data sets.

The CLIST then takes you to installation panel DSNTIPB.

3. From installation panel DSNTIPB, select the installation panel you want to update. When you finish making changes to that panel, press ENTER to return to the Update Selection Menu Panel. You can select another panel to update, or press ENTER again to complete the update process. To cancel the update session, press END.
4. Run job DSNTIJUZ

All of the installation panels can be accessed during the update process from this panel so you can view the values that you specified during installation or migration. Parameters that you can update are highlighted. Panels that do not have any updatable fields are marked with an asterisk.

```

DSNTIPB UPDATE DB2 - SELECTION MENU
===> _

Select one of the following:

 1 DATA PARAMETERS 16 OPERATOR FUNCTIONS
 2 DEFINE GROUP OR MEMBER * 17 APPLICATION PROGRAMMING DEFAULTS 1
 3 SYSTEM RESOURCE DATA SET NAMES 18 APPLICATION PROGRAMMING DEFAULTS 2
 4 DATA SET NAMES PANEL 1 19 IRLM PANEL 1
 5 DATA SET NAMES PANEL 2 * 20 IRLM PANEL 2
 6 DATA SET NAMES PANEL 3 * 21 PROTECTION
 7 DATA SET NAMES PANEL 4 * 22 MVS PARMLIB UPDATES *
 8 DATA SET NAMES PANEL 5 * 23 ACTIVE LOG DATA SET PARAMETERS
 9 CICS RELEASE * 24 ARCHIVE LOG DATA SET PARAMETERS
10 CICS DATASETS * 25 DATABASES TO START AUTOMATICALLY
11 SIZES * 26 DISTRIBUTED DATA FACILITY PANEL 1
12 THREAD MANAGEMENT 27 DISTRIBUTED DATA FACILITY PANEL 2
13 BUFFER POOL SIZES PANEL 1 * 28 STORED PROCEDURES PARAMETERS
14 BUFFER POOL SIZES PANEL 2 * 29 DATA DEFINITION CONTROL SUPPORT
15 TRACING 30 JOB EDITING

* None of the fields on these panels can be updated.

PRESS: ENTER to select RETURN to exit HELP for more information

```

Figure 60. Individual Update Menu Panel: DSNTIPB

On the command line, enter a number from 1 to 30 to select the family of parameters you wish to update. These numbers correspond to the installation panels in Table 59.

Table 59. Panel Identifiers

| Panel ID    | Panel Title                                 | Page  |
|-------------|---------------------------------------------|-------|
| 1. DSNTIPA2 | Data Parameters                             | 2-105 |
| 2. DSNTIPK  | Define Group or Member                      | 2-110 |
| 3. DSNTIPH  | System Resource Data Set Names              | 2-112 |
| 4. DSNTIPT  | Data Set Names Panel 1                      | 2-116 |
| 5. DSNTIPU  | Data Set Names Panel 2                      | 2-121 |
| 6. DSNTIPQ  | Data Set Names Panel 3                      | 2-127 |
| 7. DSNTIPG  | Data Set Names Panel 4                      | 2-130 |
| 8. DSNTIPW  | Data Set Names Panel 5                      | 2-133 |
| 9. DSNTIPV  | CICS Release                                | 2-135 |
| 10. DSNTIP3 | CICS Data Sets                              | 2-136 |
| 11. DSNTIPD | Sizes                                       | 2-139 |
| 12. DSNTIPE | Thread Management                           | 2-145 |
| 13. DSNTIP1 | Buffer Pool Sizes Panel 1                   | 2-150 |
| 14. DSNTIP2 | Buffer Pool Sizes Panel 2                   | 2-152 |
| 15. DSNTIPN | Tracing                                     | 2-154 |
| 16. DSNTIPO | Operator Functions                          | 2-158 |
| 17. DSNTIPF | Application Programming Defaults Panel 1    | 2-164 |
| 18. DSNTIP4 | Application Programming Defaults Panel 2    | 2-170 |
| 19. DSNTIPI | IRLM Panel 1                                | 2-174 |
| 20. DSNTIPJ | IRLM Panel 2                                | 2-177 |
| 21. DSNTIPP | Protection                                  | 2-181 |
| 22. DSNTIPM | MVS PARMLIB Updates                         | 2-186 |
| 23. DSNTIPL | Active Log Data Set Parameters              | 2-190 |
| 24. DSNTIPA | Archive Log Data Set Parameters             | 2-193 |
| 25. DSNTIPS | Databases and Spaces to Start Automatically | 2-199 |
| 26. DSNTIPR | Distributed Data Facility Panel 1           | 2-201 |
| 27. DSNTIP5 | Distributed Data Facility Panel 2           | 2-205 |
| 28. DSNTIPX | Stored Procedures Parameters                | 2-207 |
| 29. DSNTIPZ | Data Definition Control Support             | 2-209 |
| 30. DSNTIPY | Job Editing                                 | 2-212 |

When the panel you selected is displayed, enter the new parameters; press the ENTER key to return to the Update Selection Menu Panel. Make another panel selection or press ENTER again to process. Press END to leave the Update Selection Menu Panel and return to the Main Panel.

## Updating Other Parameters

The following methods modify some of the parameters that you cannot update through the panels:

- To update the CATALOG ALIAS and DEFINE CATALOG fields on DSNTIPA2, see Section 2 (Volume 1) of *Administration Guide*. The CATALOG ALIAS parameter establishes an alias name for your integrated catalog facility catalog. This name is also used as the high-level qualifier name for DB2 VSAM data sets. The DEFINE CATALOG parameter controls the creation of the integrated catalog facility catalog.
- To UPDATE DB2 to use the distributed data facility (DDF), follow these steps:
  1. Go through the normal UPDATE process of running the CLIST to add DDF information to installation panel DSNTIPR.
  2. Run job DSNTIJUZ.
  3. Populate the CDB. See “Step 4: Populate the Communications Database” on page 3-20.
  4. Stop and Start DB2.
  5. Bind or rebind these plans:

```
 BIND PLAN(DSNESPCS) PKLIST(*.DSNESPCS.DSNESM68)
 ISOLATION(CS) ACTION(REPLACE)
 BIND PLAN(DSNESPRR) PKLIST(*.DSNESPRR.DSNESM68)
 ISOLATION(RR) ACTION(REPLACE)
```
  6. Start DDF if you specified COMMAND instead of AUTO as the DDF STARTUP OPTION on installation panel DSNTIPR.
- To change the data set sizes for the DB2 catalog and directory:
  1. Copy the catalog and directory table spaces.
  2. Stop the table spaces or their databases.
  3. Delete the data sets and redefine them, using VSAM commands.
  4. Use the RECOVER utility to recover the catalog and directory to the new data sets.
  5. Start the table spaces or databases again.
- To change from single to dual logging for the active log:
  1. Define the second copy of the log with a VSAM IDCAMS DEFINE statement. Refer to job DSNTIJIN, which contains the DEFINE statement for the first copy of the log.
  2. Run the DSNJU003 (Change Log Inventory) utility. This adds the second copy of the log to the BSDS.
  3. Update the NUMBER OF COPIES field on installation panel DSNTIPH from 1 to 2.
  4. Run job DSNTIJUZ to make the change effective.
- To move or expand the boot strap data sets, use the IMPORT and EXPORT commands of access method service. The bootstrap data sets are accessed using JCL when DB2 starts.

#  
#  
#

- To access the log data sets, you can use stand-alone access macros or the IMPORT and EXPORT commands of access method service. See Section 4 (Volume 1) of *Administration Guide* for more information.
- To change the number of data sets for active logs, you can use the DSNJU003 utility. See Section 4 (Volume 1) of *Administration Guide* for details on the system programmer action in recovery scenarios for active log failures.





---

## Chapter 2-6. Installing the DB2 Subsystem

This chapter describes the jobs you run to install DB2, how to connect the facilities that allow TSO, batch, IMS, and CICS to access DB2 resources, and how to prepare DB2 for use.

Before you begin, you must perform SMP/E steps 1-12. They are described beginning on page 2-69. You must also run the installation CLIST. It is described beginning on page 2-89.

Before proceeding with the installation steps, refer to *IBM DATABASE 2 Program Directory* shipped with the product for keyword specifications for Preventive Service Planning (PSP). Use Information/Access or the ServiceLink facility of IBMLink to check the most current information about DB2 and other products. Contact the IBM Support Center if you do not have access to IBMLink.

You must not use secondary authorization IDs to perform any of the following installation steps.

---

### Installation Step 1: Define DB2 to MVS: DSNTIJMV

This job carries out some of the steps required to identify DB2 to MVS. This includes updating members of SYS1.PARMLIB and SYS1.PROCLIB. These data sets are documented in *OS/390 MVS Initialization and Tuning Guide*.

If job DSNTIJMV runs successfully, it produces return codes of 0.

**MVS Requirements:** Each DB2 and each IRLM you define to MVS in the IEFSSNxx parmlib member requires an MVS system linkage index (LX). The default number of these indexes that MVS reserves is 55. If you place all of your DB2 and IRLM subsystem definitions in a single IEFSSNxx member, you might need more than 55 LXs, otherwise your subsystems might not start. If you need more than 55 LXs, use the NSYSLX option on the MVS IEASYSxx parmlib member to increase this number. See *OS/390 MVS Initialization and Tuning Guide* for more information.

You must have the prerequisite level of MVS installed. Do not overwrite the MVS-supplied entries for DB2 and IRLM in the MVS program properties tables (PPT).

The PPT must contain entries for modules DSNYASCP, DXRRLM00, and DSNUTILB. MVS supplies default values for those modules. If you have modified or deleted the default values, you must enter the original values in the PPT by modifying SYS1.PARMLIB member SCHEDxx. Refer to the diagram of the PPT entry in *MVS/ESA Using the Subsystem Interface*. Use the following parameters for DSNYASCP, DXRRLM00, and DSNUTILB:

Table 60. Parameters for DSNYASCP, DXRRLM00, and DSNUTILB

| Entries  | Parameters |        |        |        |     |      |        |           |
|----------|------------|--------|--------|--------|-----|------|--------|-----------|
| DSNYASCP | CANCEL     | KEY(7) | NOSWAP | NOPRIV | DSI | PASS | SYST   | AFF(NONE) |
| DXRRLM00 | CANCEL     | KEY(7) | NOSWAP | NOPRIV | DSI | PASS | SYST   | AFF(NONE) |
| DSNUTILB | CANCEL     | KEY(7) | SWAP   | NOPRIV | DSI | PASS | NOSYST | AFF(NONE) |

**IRLM Requirements:** For later diagnosis of IRLM problems, also ensure that:

- The IRLM dump formatting module name is in control table BLSCECT in SYS1.PARMLIB.
- Load modules DXRRL186 and DXRRLFTB, and the print dump formatting module DXRRLM50 is in link list , or the job that prints the dump contains a JOBLIB or STEPLIB statement specifying the library containing the modules.

**Additional Changes to SYS1.PARMLIB and SYS1.PROCLIB:** Because different sites have different requirements for identifying DB2 to MVS, it is not possible for job DSNTIJMV to anticipate all the updates necessary. For this reason, the updates that job DSNTIJMV makes to SYS1.PARMLIB and SYS1.PROCLIB might be incomplete. You could have additional procedures of your own to rename. You can complete these updates either by making the updates directly in SYS1.PARMLIB and SYS1.PROCLIB, or by editing DSNTIJMV.

We recommend that you edit the updates directly in SYS1.PARMLIB instead of submitting the updates in the DSNTIJMV step. For SYS1.PROCLIB, submit the procedure-update section of job DSNTIJMV. Before you make the updates, read the following information and examine job DSNTIJMV to study the updates that it makes. Then use an editor such as ISPF/PDF to make the updates to SYS1.PARMLIB.

## DSNTIJMV Updates to SYS1.PARMLIB

Job DSNTIJMV updates the following SYS1.PARMLIB members:

- IEFSSNxx

This member contains an entry for every MVS subsystem. DB2 adds to this list of entries, making one entry for DB2 and two entries for the IRLM. The second IRLM entry, whose subsystem name is JRLM, is there to make it easier to add a second IRLM to your system if the first is damaged. You must provide your own procedure to add JRLM. Unique names must be used for each entry.

MVS provides subsystem entries for DB2 and IRLM in IEFSSN00. Examine these entries to determine whether they are appropriate for your needs. Make sure that a subsystem name appears only once in the subsystem name list.

You must make sure that the line describing the JES subsystem is the first line in an IEFSSNxx member. The DB2 line can come anywhere after this entry.

The DB2 entry has the following format:

```
ssname,DSN3INI,'DSN3EPX,prefix<,scope<,group-attach>>'
```

where:

*ssname*

The DB2 subsystem name.

### **DSN3INI**

is the name of the DB2 load module MVS invokes during master scheduler initialization. This module must be located in a link list data set (or in SYS1.LINKLIB).

### **DSN3EPX**

is the name of the DB2 load module that responds to DB2 requests that are received from the MVS subsystem interface. (DB2 can be

active or inactive when the requests are received.) This module must be located in a link list data set (or in SYS1.LINKLIB).

*prefix* The 1- to 8-character command prefix.

The first character of the command prefix must be one of the following: @ \$ # . / ' ) \* + - = ¢ < | & ! ; % \_ ? : ". The remaining characters of the command prefix must be one of the above characters, A-Z, or 0-9. See Table 56 on page 2-187 more information. Do not use the JES2 backspace character or command prefix character. The default is the hyphen (-).

Do not assign a command prefix that is used by another subsystem or that can be interpreted as belonging to more than one subsystem or MVS application. Specifically, do not specify a multiple-character command prefix that is a subset or a superset of another command prefix beginning from the first character. For example, it is invalid to assign '-' to one subsystem and '-DB2A' to another. Similarly, it is also invalid to assign '?DB2' to one subsystem and '?DB2A' to another. It is valid to assign '-DB2A' and '-DB2B' to different DB2 subsystems.

*scope* The 1-character scope for the command prefix. DB2 registers its command prefix with MVS. When this is done, the scope of the command prefix is controlled by the value you choose:

S Started, and register the prefix with Sysplex scope at DB2 startup instead of during MVS IPL. This is the default.

**Recommendation:** Choose S, which allows you to have a single IEFSSNxx parmlib member to be used by all MVS systems in the Sysplex. It also simplifies the task of moving a DB2 from one system to another; you can stop DB2 on one MVS and start it up on another. There is no need to re-IPL the system.

M MVS system scope, and register the prefix during MVS IPL.

X Sysplex scope, and register the prefix during MVS IPL. This means this DB2 cannot be restarted on another MVS without changing the definitions and re-IPLing both MVSs.

For more information about the command prefix facility of MVS, see *OS/390 MVS Planning: Operations*.

*group-attach*

The group attachment name, used for data sharing. You can specify this on installation panel DSNTIPK.

- IEAAPFxx or PROGxx

Job DSNTIJMV updates IEAAPFxx to include the DB2 program libraries (*prefix*.SDSNEXIT, *prefix*.SDSNLOAD, *prefix*.SDXRRESL, and *prefix*.SDSNLINK) as APF-authorized libraries.

If the program library containing DFSORT is not already APF-authorized, you can edit DSNTIJMV to authorize it. To do this, you can include the authorization either in this list or in LNKLSTxx. All libraries concatenated with *prefix*.SDSNLOAD in STEPLIB and JOBLIB statements must be APF-authorized. Be sure that the volume serial number in this member is the volume on which the data set resides.

If you are using MVS/ESA Version 4 Release 3, you might be using the PROGxx member instead of the IEAAPFxx member. If so, you have to update this member manually—job DSNTIJMV does not edit it.

- LNKLSTxx

Whether you edit the updates directly or edit DSNTIJMV to make the updates, you might first want to review “Choosing Link List Options” on page 2-74.

Job DSNTIJMV updates this member to include the DB2 load module library, *prefix*.SDSNLINK, in the LNKLSTxx. If you moved the modules from *prefix*.SDSNLINK into another library, edit DSNTIJMV to include that library in the LNKLSTxx. If you have combined *prefix*.SDSNLINK and *prefix*.SDSNLOAD into one library, edit DSNTIJMV to include the combined library in the LNKLSTxx. See *OS/390 MVS Initialization and Tuning Guide* for restrictions on data sets that are concatenated in LNKLST.

#  
#

Any data set that is added to the LNKLSTxx member must be cataloged in the master catalog of the system. This is normally true of *prefix*.SDSNLINK; however, if an alias points to a user catalog when you run DSNTIJAE, *prefix*.SDSNLINK is cataloged in a user catalog. In this case, you must either ensure that *prefix*.SDSNLINK is also cataloged in the master catalog or give *prefix*.SDSNLINK a high-level qualifier other than *prefix*, the high-level qualifier for this release. You must give a high-level qualifier other than *prefix* to all release-sensitive data sets placed in the LNKLSTxx member.

If you do not include the DFSORT library in the LNKLSTxx member, you must provide a JOBLIB or STEPLIB statement for all utility jobs that include the DFSORT program library. You can accomplish this by placing a STEPLIB statement in DSNUPROC, which appears later in this job. If you use customized modules and exits, *prefix*.SDSNEXIT must precede *prefix*.SDSNLOAD in JOBLIB and STEPLIB statements.

|  
|  
|

You must do additional editing for the SYS1.PARMLIB updates. If you are editing DSNTIJMV, rather than making the changes directly, you have a choice: either include your additional entries for the SYS1.PARMLIB members (IEAAPFxx and LNKLSTxx) at the end of the existing list of entries, or place them earlier in the list.

If you include them at the end of the existing SYS1.PARMLIB entries, make sure there are commas (the continuation character) delimiting each entry except the last.

Another SYS1.PARMLIB change to consider at this time is the extended common storage area (ECSA) size, specified in the CSA parameter of the IEASYS00 parameter. Be sure that you have specified an adequate size for this subsystem (generally 2MB plus the MAXIMUM ECSA on installation panel DSNTIPJ if the CROSS MEMORY value is NO).

The **IOP parameter** is another SYS1.PARMLIB change to consider at this time. If you are running with MVS/SP Version 4 Release 3 or later and DFSMS/MVS Version 1 Release 1, DB2 can schedule I/O priority. To enable this, you must:

- Use the IOP parameter to set the I/O priority for the address space of a performance group. The IOP parameter is in the IEAIPSxx member of SYS1.PARMLIB.
- Enable MVS I/O priority scheduling by specifying IOQ=PRTY in the IEAIPSxx member of SYS1.PARMLIB.

You must issue an IPL command for MVS for the PARMLIB updates to take effect. To avoid issuing an IPL command for MVS during DB2 installation, you can make these updates and issue the IPL well in advance of your DB2 installation or migration session. See Section 5 of *Administration Guide* for more information on I/O priority scheduling.

## DSNTIJMV Updates to SYS1.PROCLIB

Job DSNTIJMV updates SYS1.PROCLIB to include the DB2 procedures. The procedure names must begin with xxxx, the subsystem name, and must end with either MSTR, DBM1, or DIST.

- System services address space startup procedure (xxxxMSTR)
- Database services address space startup procedure (xxxxDBM1)
- Distributed data facility address space startup procedure (xxxxDIST)
- Stored procedures address space (xxxxSPAS or any user-defined address space name)
- IRLM address space startup procedure (IRLMPROC or user-defined address space name)
- Precompiler procedures
- Utilities procedure (DSNUPROC).

Examine the SYS1.PROCLIB updates carefully. You might want to use a procedure library other than SYS1.PROCLIB for the procedures. Four of the procedures are used for startup tasks; the other procedures are used to prepare application programs for execution and to invoke DB2 utilities. The program preparation procedures are required for the sample applications and can be helpful in generating other JCL procedures.

Change any data set names that differ at your site. If you specified a suffix on panel DSNTIPA1, that suffix is appended to data sets &USER..DBRMLIB.DATA.suffix, &USER..RUNLIB.LOAD.suffix, and &USER..SRCLIB.DATA.suffix. To override these data set names, you must edit the updates to SYS1.PROCLIB.

The language preparation procedures in job DSNTIJMV use the DISP=OLD parameter to enforce data integrity. However, when the installation CLIST is executed, the DISP=OLD parameter for the DBRM library data set is modified to DISP=SHR. This could cause data integrity problems when you run multiple precompiler jobs. To avoid these data integrity problems, if you are not using DFSMSdfp's partitioned data set extended (PDSE), you must change the language preparation procedures (DSNHCOB, DSNHCOB2, DSNHICOB, DSNHICB2, DSNHFOR, DSNHC, DSNHCPP, DSNHCPP2, DSNHPLI, DSNHASHM) to specify the DISP=OLD parameter instead of the DISP=SHR parameter.

If compiler STEPLIB statements are needed, add them.

Examine the size of the private area on the DB2 start procedures. If necessary, modify the procedures to satisfy the requirements for environmental descriptor manager (EDM) pool size, buffers, number of data sets open, and the amount of available private address space. For more information about private address spaces, refer to "Working Storage Calculation" on page 2-46.

---

## Installation Step 2: Define the Integrated Catalog Facility Catalog and Alias: DSNTIJCA

The integrated catalog facility catalog is the VSAM object in which DB2 catalogs the data sets you create during the process of installing. Job DSNTIJCA creates the integrated catalog facility catalog and its alias. DB2 uses the catalog alias as the prefix for your DB2 VSAM data sets.

Running DSNTIJCA is optional. If you specified YES for the DEFINE CATALOG option on installation panel DSNTIPA2, you must run this job to create the catalog. Before running this job, examine the DEFINE UCAT statement carefully to be sure that the parameters are appropriate for your needs.

Do not run this job if you want to use an existing integrated catalog facility catalog and alias (that is, you specified NO for the DEFINE CATALOG parameter on installation panel DSNTIPA2). However, make sure that the integrated catalog facility catalog you are going to use is created and that you defined an integrated catalog facility catalog alias.

If job DSNTIJCA runs successfully, it produces return codes of 0. If DSNTIJCA fails or abends, delete the integrated catalog facility catalog (if it was created) and rerun the job. To delete the integrated catalog facility catalog, run job DSNTIJDE (see page 2-238 below).

**Deleting DB2 Data Sets: DSNTIJDE:** Job DSNTIJDE is not part of the normal installation process; use this job only for rerunning part of the process. Do not run this job during migration or fallback.

This job deletes the previously created data sets for the DB2 directory and DB2 catalog. If a job fails or abends, you might need to run this job before restarting the DB2 installation process.

In most cases, you must remove or comment out the delete statement in this job for the integrated catalog facility catalog (if the statement is present). It is likely that the integrated catalog facility catalog does not need to be deleted and redefined.

Deletes might fail for data sets that do not exist. This does not necessarily indicate that the job failed. If you receive other messages, check them carefully.

Job DSNTIJDE does not work properly if job DSNTIJSG has been executed. This job does not delete the resource limit specification table or the data sets used by the distributed data facility.

If job DSNTIJDE fails or abends, correct the error conditions and rerun the job. If you want to delete the integrated catalog facility catalog, first list its contents and delete the data sets cataloged there. This could include sample data sets, user-defined data sets, or subsystem data sets that were not deleted properly. You can use a FORCE command to delete the user catalog.

If you delete the catalog using FORCE before deleting all the data sets, you can use the RECATALOG option of DEFINE CLUSTER and delete the data sets.

---

## Installation Step 3: Define System Data Sets: DSNTIJIN

Job DSNTIJIN defines VSAM and non-VSAM data sets for DB2. It does the following:

- Defines three non-VSAM data sets for the DB2 sample objects:

*prefix*.DBRMLIB.DATA  
*prefix*.RUNLIB.LOAD  
*prefix*.SRCLIB.DATA

- Defines the VSAM clusters for the bootstrap data sets.

Each bootstrap data set (BSDS) consists of a VSAM key-sequenced data set. You defined the BSDS names during the ISPF tailoring session.

- Defines the VSAM clusters for the active log data sets.

You specified up to 31 primary active log data sets during the ISPF tailoring session (NUMBER OF LOGS on installation panel DSNTIPL). You might also have requested dual logging to generate two copies of each active log data set. Consequently, job DSNTIJIN can define up to 62 active log data sets.

- Defines the DB2 directory database.

Job DSNTIJIN creates and catalogs the DB2 directory database (DSNDB01). The DB2 directory database contains information that is required to start DB2 and is also used by DB2 during its normal operation. It contains table spaces and index spaces that the installation job allocates.

- Defines the DB2 catalog database.

Job DSNTIJIN creates and catalogs the DB2 catalog database (DSNDB06). The DB2 catalog contains information about every object that DB2 maintains.

- Invokes the LISTCAT command of access method service so you can check that the VSAM definitions were successful.

Check the DEFINE CLUSTER statements in job DSNTIJIN to ensure that they allocate adequate DASD for your system. See Section 5 (Volume 2) of *Administration Guide* for guidance on allocating and extending data sets. Also, for recovery purposes, it is best to place system data sets like the DB2 recovery log and the VSAM catalog on different DASD volumes. Because these data sets are used frequently, do not migrate them with DFSMSHsm.

If DSNTIJIN runs successfully, it produces return codes of 0 for all DEFINE statements and steps. Check any VSAM messages carefully.

If job DSNTIJIN fails or abends, remove the MVS catalog delete statements from job DSNTIJDE, run DSNTIJDE (to delete the data sets created by DSNTIJIN), and rerun DSNTIJIN. For information about job DSNTIJDE, see page 2-238.

---

## Installation Step 4: Initialize System Data Sets: DSNTIJID

Job DSNTIJID initializes VSAM data sets for DB2. It performs these functions:

- Initializes the BSDS by invoking the change log inventory utility.
- Initializes the DB2 directory database using data in the untailed version of *prefix*.SDSNSAMP.

- Initializes the DB2 catalog database with data from the untailed *prefix.SDSNSAMP*.
- Invokes the DSNJLOGF utility to preformat active log data sets that are created during installation. See *Utility Guide and Reference* for more information on DSNJLOGF.

If job DSNTIJID runs successfully, it produces return codes of 0. If you receive any VSAM messages, check them carefully. If DSNTIJID fails or abends, run job DSNTIJDE, then rerun DSNTIJIN and DSNTIJID .

---

## Installation Step 5: Define DB2 Initialization Parameters: DSNTIJUZ

Job DSNTIJUZ generates the DB2 subsystem parameter module DSNZPARM (or the name you specified for PARAMETER MODULE on installation panel DSNTIPO) and the data-only load module DSNHDECP.

Seven macros expand to form the data-only subsystem parameter load module. It contains the DB2 execution-time parameters that you selected using the ISPF panels. These seven macros are DSN6ARVP, DSN6ENV, DSN6FAC, DSN6LOGP, DSN6SPRM, DSN6SYSP, and DSN6GRP.

The DSNTINST CLIST performs calculations on some of the parameter values you enter during an installation session. The results of these calculations appear in the macro descriptions.

Besides defining subsystem parameters and DSNHDECP, job DSNTIJUZ does the following:

- Link-edits the assembled modules into the *prefix.SDSNEXIT* library.
- Places archive log data set and system database passwords that you specify into the BSDS. It also updates the BSDS with DDF information with the change log inventory utility. In a data sharing environment, data sharing information is updated too.
- Uses the assembler, the DSNHDECM macro, and SMP/E to move your subsystem name and the values you specified on installation panels DSNTIPF and DSNTIP4 into another data-only load module called DSNHDECP. DSNHDECP also contains the default SSID from the SUBSYSTEM NAME field on installation panel DSNTIPM. Job DSNTIJUZ includes DSNHDECP in various DB2 load modules.
- Uses SMP/E in step DSNTIMQ to read in the edited version of DSNTIJUZ. This is required to pick up the appropriate includes and library names. After the initial run of step DSNTIMQ, it is only required when changes have been made to DSNHDECP.
- Uses JCLIN to ensure that DSNHDECP service is placed in all the required load modules.

APAR PQ03621 modified subsystem parameter ABIND, in macro DSN6SPRM to accept an additional value of COEXIST. To use this value, you must manually specify COEXIST for macro DSN6SPRM ABIND in job DSNTIJUZ. For details on when to use COEXIST, see 2-160.



# A new subsystem parameter PARTKEYU, has been added to macro DSN6SPRM.  
 # This parameter specifies whether values in columns that participate in partitioning  
 # keys may be updated. A value of YES means the values in such columns may be  
 # updated. The default value is YES. A value of NO means the values in such  
 # columns may not be updated. A value of SAME means the values in such columns  
 # may be updated if and only if the update leaves the updated row in the same  
 # partition as the original row.

# A new subsystem parameter PARAMDEG, has been added to macro DSN6SPRM.  
 # This parameter specifies the maximum degree of parallelism for a parallel group.  
 # The default value of 0 means there is no upper limit for the amount of parallelism.  
 # A non-zero value (1-254) indicates the maximum degree of parallelism for this  
 # parallel group.

# A new subsystem parameter EDMBFIT, has been added to macro DSN6SPRM.  
 # This parameter is used to adjust the free chain search algorithm on systems with a  
 # large EDM pool (greater than 40M). The default value of NO means use a first fit  
 # algorithm. The YES value means use a better fit algorithm.

# New subsystem parameters added after the initial availability of Version 5 need to  
 # be manually added or changed in job DSNTIJUZ. The new parameters are  
 # described in "Subsystem Parameters added after General Availability" on  
 # page 2-226.

| If you added a STEPLIB statement to the DB2 start procedures, modify the  
 | SYSLMOD steps to point to the library in the STEPLIB statement instead of the one  
 | in *prefix.SDSNEXIT*.

If the DB2 distribution library prefix is different from the target library prefix, edit  
 DSNTIJUZ to correct the data set name for *prefix.ADSNLOAD*.

If you have not run the SMP/E ACCEPT job (DSNTIJAC) of FMID HDB5510, you  
 must edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included  
 in the concatenation for the ADSNLOAD DD statement in steps DSNTIZL and  
 DSNTIZQ.

# You might receive message GIM65001 when running steps DSNTLOG and  
 # DSNTIMQ, or you might receive a return code of 4 when running step DSNTIMQ.  
 You can ignore these messages.

If job DSNTIJUZ fails or abends, correct the problem and rerun the job.

---

## Installation Step 6: Define User Authorization Exit Routines: DSNTIJEX (Optional)

Job DSNTIJEX builds the sample authorization exits, DSN3@SGN and DSN3@ATH, from the source code in *prefix.SDSNSAMP* and places them in the *prefix.SDSNEXIT* library. The DB2 CLIST tailors the JCL in DSNTIJEX to match your site's environment.

The sample authorization exits are not the same as the default authorization exits supplied by DB2. By implementing the sample authorization exits you can provide group names as secondary authorization IDs. By modifying the sample authorization exits, you can tailor authorization processing for your subsystem. For

information on writing exit routines, see Appendix B (Volume 2) of *Administration Guide* . For more information on controlling data access, see Section 3 (Volume 1) of *Administration Guide* .

You have the following options regarding exit routines:

- To use the sample authorization exits, run job DSNTIJEX.
- To use the default authorization exits, skip job DSNTIJEX.
- To use the modified sample authorization exits, modify DSNTIJEX to reference the correct library before you run it.

If job DSNTIJEX runs successfully, it produces return codes of 0.

If job DSNTIJEX fails or abends, correct the problem and rerun the job.

---

## Installation Step 7: Record DB2 Data to SMF (Optional)

When you install DB2, you can specify if DB2 statistical, accounting, and audit trace data are to be collected.

- To have DB2 collect statistical information, accept the default (YES class 1) for the SMF STATISTICS option on installation panel DSNTIPN. To collect statistical information for deadlock or timeout, specify class 3. To collect information about DDF error conditions, specify class 4.
- To have DB2 collect accounting information, accept the default (1) or specify \* (all classes) for the SMF ACCOUNTING option on installation panel DSNTIPN. You must consider which classes should be turned on.
- To have DB2 collect auditing information, specify \* (all classes) for the AUDIT TRACE option on installation panel DSNTIPN. You must consider which classes should be turned on.

For more information on the START TRACE command, see Chapter 2 of *Command Reference*. In all cases, DB2 invokes a trace, passing the data it collects to the system management facility (SMF) of MVS.

DB2 also passes performance data to SMF whenever an accounting, statistics, or audit trace is successfully started or stopped. This is not the only performance data that DB2 can record. After you complete the installation process, you can use commands to have DB2 record performance data for over 230 different subsystem events.

See Chapter 2 of *Command Reference* to see what kind of data DB2 can collect and pass to SMF.

You must make some additional updates if, during installation, you requested that DB2 pass accounting and statistics data to SMF. Specifically, you must update the SMFPRMxx member of SYS1.PARMLIB as follows:

- Specify the ACTIVE parameter
- Specify the proper TYPE subparameter of SYS and SUBSYS

During DB2 execution, you can use the SMF SET or SS command to alter the SMF parameters. For example, the following command allows you to record the statistics trace class 1 IFCIDs 0001, 0002, and 0202 (SMF record type 100); accounting

trace class 1 IFCIDs 0003 and 0239 (SMF record type 101); and all other DB2 trace records (SMF record type 102) to SMF:

```
SYS(TYPE(100:102))
```

If you have DB2 pass data to SMF, you must allocate an adequate supply of SMF buffers. The default buffer settings are probably insufficient.

You can specify SMF buffering on the VSAM BUFSP parameter of the access method services DEFINE CLUSTER statement. Do not use the default settings if DB2 data is sent to SMF. Specify CISZ(4096) and BUFSP(81920) on the DEFINE CLUSTER statement for each SMF VSAM data set. These values for CISZ and BUFSP are the minimum requirement for DB2. You might need higher values for CISZ and BUFSP depending on the requirements of all your MVS subsystems.

You can also code an IEFU84 SMF exit to process the records that are produced.

Detailed information about starting and stopping DB2 traces for performance, accounting, audit, and statistics data is presented in Section 5 (Volume 2) of *Administration Guide*.

For more information on SMF, refer to *MVS/ESA Using the Subsystem Interface* and *OS/390 MVS Initialization and Tuning Guide*.

---

## Installation Step 8: Establish Subsystem Security (Optional)

DB2 includes means for controlling access to data within DB2. It also works together with outside security systems, such as RACF, that control access to the DB2 system. See Section 3 (Volume 1) of *Administration Guide* for suggestions and instructions for including DB2 in your security system.

---

## Installation Step 9: Connect DB2 to TSO

Although you can eventually connect DB2 to IMS, CICS, or both, we suggest you connect only TSO at first. At this point you can run the sample applications that do not require CICS or IMS, allowing your database and system administrators to gain familiarity with the administrative facilities of DB2 Version 5.

If you have previously installed DB2 and are performing that task again, your database and system administrators are probably already familiar with DB2. In this case, you can connect IMS, CICS, or both at the same time you connect TSO. You can then run the sample applications that require CICS and IMS at the same time you run the sample applications for TSO and batch.

To attach DB2 to TSO, you must do the following:

1. Make DB2 load modules available to TSO and batch users.
2. Make DB2 CLISTs available to TSO and batch users.
3. Make PL/I options available (if applicable).
4. Make panels, messages, and load modules available to ISPF and TSO.
5. Connect the DB2I panels to the ISPF Main Panel.
6. Establish TSO and RACF user IDs for DB2 users.

These tasks are discussed in the following sections.

## Making DB2 Load Modules Available to TSO and Batch Users

If you included *prefix*.SDSNEXIT and *prefix*.SDSNLOAD in your LNKLISTxx, you can skip this step.

If you have not included *prefix*.SDSNEXIT and *prefix*.SDSNLOAD in your LNKLISTxx, you must add STEPLIB statements to your logon procedures and JCL for jobs to ensure that you access the DB2 Version 5 load modules.

If *prefix*.SDSNEXIT is not in your LINKxx, then add it to your STEPLIB and JOBLIB concatenations before *prefix*.SDSNLOAD. Refer to “Choosing Link List Options” on page 2-74 for information on link lists.

## Making DB2 CLISTs Available to TSO and Batch Users: DSNTIJVC

From *prefix*.SDSNCLST, the DSNTINST CLIST reads and edits these four CLISTs: DSNEMC01, DSNH, DSNU, and DSNHC. It then places those CLISTs in *prefix*.NEW.SDSNTEMP. You might want to modify the default values. See “Completing the CLIST Processing” on page 2-219 for information on the items you want to modify. The DSNEMC01 CLIST is only executed the first time a TSO user runs any of the DB2I functions. The ISPF profile member DSNEPROF has its variables set by DSNEMC01 according to the values specified on installation panel DSNTIPF.

Job DSNTIJVC merges the tailored CLISTs from *prefix*.NEW.SDSNTEMP with unchanged CLISTs from *prefix*.SDSNCLST, and places all CLISTs in *prefix*.NEW.SDSNCLST. It also converts the record format of the DB2 CLISTs from fixed block to variable block with a record length of 84 and a block size of 3120.

**If you use fixed-block format CLIST libraries**, modify job DSNTIJVC as follows:

- Change the SYSIN DD statement to DUMMY.
- Change the allocation of *prefix*.NEW.SDSNCLST to match the data control block (DCB) attributes of your other CLIST libraries.

A CLIST that has been converted from fixed block to variable block cannot be used as input to the DSNTINST CLIST; use the unedited version of the SDSNCLST data set, as created by SMP.

To make the CLISTs available to TSO and batch users, you must either concatenate *prefix*.NEW.SDSNCLST with your existing CLIST libraries or copy *prefix*.NEW.SDSNCLST into an existing CLIST library.

If you need to rerun this job, first delete data set *prefix*.NEW.SDSNCLST, which is created by this job.

When corrective service is applied to a CLIST, SMP/E changes only the *prefix*.SDSNCLST data set. You need to redo any record format changes and reapply any needed tailoring. You also need to move the CLIST to *prefix*.NEW.SDSNCLST. Corrective service (program temporary fixes) for these CLISTs is sent with ++HOLD statements, noting that this additional work might be required.

## Ensuring that PL/I Options are Available

If you are using PL/I, ensure that the options your DB2 programmers use are included in the compiler. Restrictions imposed by your site on PL/I compiler options affect how you can use DB2 program preparation. The program preparation function uses the following options:

FLAG OBJECT SOURCE TERMINAL XREF

If the macro pass is used, the following options are needed as well:

MACRO MDECK SYNTAX

## Making Panels, Messages, and Load Modules Available to ISPF and TSO

You must concatenate the DB2 ISPF libraries with the ISPLIB, ISPSLIB, and ISPMLIB DD statements in your logon procedures and in any of your CLISTs where they might be allocated. These libraries are *prefix.SDSNSPFP*, *prefix.SDSNSPFM*, *prefix.SDSNSPFS*, and either *prefix.SDSNPFPE* or *prefix.SDSNPFPK* depending on whether you are using English or Kanji DB2I panels. If you are using Online Help, include *prefix.SDSNSPFT*.

DB2I uses the ISPF PROFILE and SHARED variable pools for most panel variable fields. This makes it easier to reenter a panel when panel variables have previously been specified. For the DB2 subcommands that permit LISTS of plan names, package names, DBRMs, and ENABLE and DISABLE statements, DB2I provides ISPF to contain all the user-specified variables for these subcommand keywords.

DB2I creates and maintains a set of ISPF tables in a user-defined TSO data set that is allocated to a ddname of DSNETBLS. The DB2I-generated tables in this library are DSNCONNS, DSNDBRMS, and DSNPLPKN. Table 61 shows the library table member names and their contents.

Table 61. The DB2 ISPF Table Library

|          |                                                                                                 |
|----------|-------------------------------------------------------------------------------------------------|
| DSNCONNS | ENABLE/DISABLE connection type and connection name variables referenced by plan or package name |
| DSNDBRMS | Subcommand DBRM member and LIBRARY name variables referenced by plan name                       |
| DSNPLPKN | Package list variables referenced by package name                                               |

When allocating this data set, the following DCB attributes must be assigned:

```
DSORG(P0) RECFM(F B) LRECL(80) BLKSIZE(n*LRECL)
```

where n is any integer.

The following example shows how you might set up an ALLOCATE statement to create the data set:

```
ALLOC DA(DSNPFT) NEW SP(1 1) TR DIR(10) +
DSORG(P0) RECFM(F B) LRECL(80) BLKSIZE(800)
F(DSNETBLS) REUSE
```

The following example shows how you might allocate an existing data set to the DSNETBLS ddname:

```
ALLOC DA(DSNSPFT) F(DSNETBLS) REUSE
```

Add the DSN command to the HELP list of available TSO commands by editing SYS1.HELP(COMMANDS), and add a line indicating that the DSN command allows you to perform DB2 functions from TSO. SYS1.HELP(COMMANDS) is HELP information only. It describes DB2 function; it does not provide that function.

```
DB2I uses ISPF table services to maintain individual ISPF tables within the
DSNETBLS data set. For performance reasons, ISPF keeps this table library in an
open state once an individual table has been updated. Attempts to close this data
set using the TSO FREE command will result in error message IKJ568611.
```

```
For additional information on this TSO error message and how to close this data
set, refer to ISPF Version 4 Messages and Codes.
```

If you want to run the ISPF/CAF sample application provided with DB2, be sure that the data set *prefix.RUNLIB.LOAD* is included in the logon procedures or in the ISPLLIB concatenation. For more information about the ISPF/CAF sample application, see “Running Dynamic SQL and the ISPF/CAF Application” on page 2-321.

## Connecting DB2I Panels to the ISPF Main Panel

This section explains how to connect the DB2 panels to the standard ISPF panels already installed on your system. We recommend using the following panels for establishing the connection. See your TSO administrator for other possibilities.

- ISP@MSTR, ISR@PRIM, or ISRFPA for the connection to DB2 Interactive services
- ISR00003 for the tutorial menu update

Two example panels are provided here. Their names are DSNTIPRM (the DB2 version of an ISPF primary options panel) and DSNTIPTU (the DB2 version of a tutorial table of contents). Using the TSO RENAME command, give DSNTIPRM an alias of ISR@PRIM, and give DSNTIPTU an alias of ISR00003. For example:

```
RENAME 'prefix.SDSNSPFP(DSNTIPRM)' (ISR@PRIM) ALIAS
RENAME 'prefix.SDSNSPFP(DSNTIPTU)' (ISR00003) ALIAS
```

If the DB2 panel library is concatenated before the standard ISPF library, the connection is made.

If your site has made changes to either of these panels, change your existing panels rather than use the following examples. The panels in Figure 61 on page 2-247 and Figure 62 on page 2-249 display the needed modifications.



DSNECPRI is a CLIST and can be invoked directly from another user CLIST. It is an alternative way to invoke DSNEPRI without updating the primary ISPF panel.

# By specifying NEWAPPL(DSNE), you define DSNE as the ISPF application name  
# used by DB2I. ISPF uses the name DSNE to create the ISPF profile pool member  
# name (DSNEPROF) in the TSO\_userid.ISPPROF data set, which will contain all  
# ISPF panel variables defined during DB2I execution.

# **Using a NEWAPPL name other than DSNE:**You may define any valid ISPF  
# application name. If you define an application name to be something other than  
# DSNE and you plan to use DB2 online help, you must create a new command table  
# member that corresponds to your application name. You may create a new  
# command table name by using the TSO Command Table Utility and specifying the  
# Verb as Help and the Action as Select Panel(&Helppan) after defining your new  
# ISPF application name.



```

)ATTR
/*****
/* COPYRIGHT = 5740-XJR (C) COPYRIGHT IBM CORP 1982, 1985, 1990 */
/* REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083 */
/* STATUS = VERSION 4, LEVEL 0 */
/*****
)BODY
%TUTORIAL ----- TABLE OF CONTENTS ----- TUTORIAL
%OPTION ==>_ZCMD

 | ISPF PROGRAM DEVELOPMENT FACILITY TUTORIAL |
 | TABLE OF CONTENTS |
 |-----|

The following topics are presented in sequence, or can be selected by
entering a one-character selection code in the option field on line 2:
%G+ GENERAL - General information about ISPF
%0+ ISPF PARMS - Specify terminal and user parameters
%1+ BROWSE - Display source data or output listings
%2+ EDIT - Create or change source data
%3+ UTILITIES - Perform utility functions
%4+ FOREGROUND - Invoke language processors in foreground
%5+ BATCH - Submit job for language processing
%6+ COMMAND - Enter TSO command or CLIST
%7+ DIALOG TEST - Perform dialog testing
%8+ DB2 - Information about DB2
%X+ EXIT - Terminate ISPF using log and list defaults
The following topics are presented only if explicitly selected:
%A+ APPENDIX A - Dynamic allocation interface routine (DAIR) errors
%B+ APPENDIX B - ISPF listing formats
%I+ INDEX - Alphabetic index of tutorial topics
)PROC
&ZSEL = TRANS(&ZCMD
 G,ISR01000
 0,ISP05000
 1,ISR10000
 2,ISR20000
 3,ISR30000
 4,ISR40000
 5,ISR50000
 6,ISR60010
 7,ISR70000
8,DSN4V2DB
 X,ISP90100
 A,*ISP93030
 B,*ISR95000
 I,*ISR91000
)
)END

```

Figure 62. ISPF Program Development Facility Tutorial Panel (DSNTIPTU), Edited to Include DB2 Tutorial

The DSNTIPTU panel is shown in Figure 62. Notice the two added lines in boldface type. Adding these lines allows you to invoke the DB2 tutorial panels. The two added lines include one displayed line:

```
%8+ DB2 - Information about DB2
```

and one undisplayed line:

```
8,DSN4V2DB
```

The displayed line presents the user with a choice for the DB2 tutorial. The undisplayed line actually invokes the DB2 tutorial menu (DSN4V2DB).

For information about ISPF, see *ISPF Version 4 for MVS Dialog Management Guide* and *ISPF/PDF Version 4 for MVS Guide and Reference*.

## Establishing DB2 Authorization IDs in TSO and RACF

You specified the following IDs on installation panel DSNTIPP:

- Two system administrator (installation SYSADM) authorization IDs
- Two system operator (installation SYSOPR) authorization IDs
- One authorization ID (installation IBMUSER) if RACF is not available for batch access and USER= is not specified in the job statement.

Before attempting to access DB2, be sure that the installation SYSADM IDs you specified are defined in your TSO and RACF systems. You also can define installation SYSOPR IDs there, as well as the installation IBMUSER ID.

---

## Installation Step 10: Connect IMS to DB2 (Optional)

Connecting DB2 to IMS requires coordination with your IMS support group. To connect the IMS attachment facility, you must:

- Make DB2 load modules available to IMS
- Define DB2 to IMS
- Define new application programs and transactions to IMS
- Prepare IMS applications for DB2

Depending on your site, you also could need to:

- Define DB2 plans for IMS applications
- Generate a user language interface.

These tasks are discussed in “Chapter 2-10. Connecting the IMS Attachment Facility” on page 2-371. This chapter also covers the requirements from a DB2 perspective and refers you to IMS books for specific IMS information.

---

## Installation Step 11: Connect CICS to DB2 (Optional)

To connect DB2 to CICS you must regenerate several CICS tables with additional entries. A macro is supplied with DB2 to define the connection between CICS and DB2 using a resource control table (RCT). Be sure that you coordinate this connection with your CICS support group. To connect the CICS attachment facility, you must do the following:

- #
- #
- Add CICS routines to DB2 load modules by running job DSNTIJSU (Job DSNTIJSU is not required for CICS Version 4 and subsequent releases.)
- Calculate space requirements for the CICS attachment facility
- Define DB2 to CICS using the RCT
- Update the CICS system tables
- Update the CICS initialization JCL
- Coordinate DB2 and CICS security
- Prepare CICS applications for DB2.

These tasks are discussed in “Chapter 2-11. Connecting the CICS Attachment Facility” on page 2-379 along with the activities required to support CICS in a DB2 environment.

If you use CICS 3.3, you must manually add the TASKDATALOC(BELOW) parameter to the transaction definition. For more information, see “Transaction Entries” on page 2-387.

---

## Installation Step 12: IPL MVS

The first time you start DB2, you must IPL MVS, then start the DB2 subsystem.

For Version 5, the load module library SDSNLINK contains the early code. SDSNLINK contains modules that must be placed in the link list look aside address space (LLA) because they are loaded at subsystem initialization during the IPL.

The MVS IPL is necessary because installation job DSNTIJMV makes changes to SYS1.PARMLIB that are not recognized by MVS until the next IPL. DSNTIJMV makes the following changes to the SYS1.PARMLIB library:

- Creates new subsystem definitions in the IEFSSNxx member
- Creates new APF libraries in the IEAAPFxx member
- Creates new load module libraries in the LNKLSTxx member.

Complete these changes before performing the IPL. For more information, refer to “Installation Step 1: Define DB2 to MVS: DSNTIJMV” on page 2-233 and “Choosing Link List Options” on page 2-74.

During the MVS IPL, message DSN3100I appears on the MVS console, stating that DB2 is ready for the START command.

---

## Installation Step 13: Start the DB2 Subsystem

Perform the following steps to start DB2:

1. Start the IRLM.

If you have not requested that DB2 automatically start the IRLM, you must start it before you start DB2. Use the command:

```
START irlmproc
```

where *irlmproc* is the name you specified for the PROC NAME option on IRLM Panel 1 (DSNTIPI).

If you specified YES for the AUTO START option on IRLM Panel 1 (DSNTIPI), DB2 starts the IRLM automatically.

2. Start DB2 from the MVS console. Use the command:

```
-DSN1 START DB2
```

where (-DSN1) is the subsystem command prefix you defined for DB2.

DB2 uses the subsystem parameter module specified in the startup JCL procedure in SYS1.PROCLIB:

```
//IEFPROC EXEC PGM=DSNYASCP,PARM='ZPARAM(DSNZPxxx)', ...
```

where *DSNZPxxx* is the value you specified for PARAMETER MODULE on panel DSNTIPO.

If you need to change the *DSNZPxxx*, you can edit SYS1.PROCLIB. Or, you can override the *DSNZPxxx* by using the PARM option as follows:

```
-DSN1 START DB2, PARM(DSNZPxxx)
```

If DB2 starts successfully, 2 to 5 address spaces also start. These address spaces are *ssnmMSTR* and *ssnmDBM1*, and possibly *ssnmDIST*, *ssnmSPAS*, and *irlmproc*, where *ssnm* is the DB2 subsystem name and *irlmproc* is the IRLM procedure name.

If DB2 starts successfully, the series of RESTART messages you receive concludes with these two messages:

```
DSNR002I RESTART COMPLETED
DSN9022I DSNYASCP '-DSN1 START DB2' NORMAL COMPLETION
```

After you start DB2, check outstanding conditions. Identify unusual conditions for databases with the command:

```
-DSN1 DISPLAY DATABASE(*) SPACENAM(*) RESTRICT
```

If DB2 does not start successfully, it usually abends with a reason code indicating where the error occurred. To find the error, check the set of definitions for the associated resource. Be sure that the DSNTIJUZ, DSNTIJIN, and DSNTIJID jobs ran correctly. Also, check that the subsystem parameter member you specified (or allowed to default) when you started DB2 is the one built by job DSNTIJUZ. Check the JCL for the DB2 startup procedure.

**Note to distributed data facility users:**

VTAM must be defined before DDF can start. But, you do not need to have TCP/IP configured to start DDF.

3. Optionally, start TSO.

If you want to use the TSO SUBMIT command to do housekeeping and installation verification, you must start TSO (if it is not already started).

---

## Installation Step 14: Define Temporary Work Files: DSNTIJTM

The DSNTIJTM job defines the database for temporary work files and provides some cleanup. For non-data-sharing installations, the work file database is DSNDDB07. This job assembles, link-edits, binds, and runs DSNTIAD, a program that processes certain SQL statements dynamically. It also defines the initial buffer pool and hiperpool sizes as specified on installation panels DSNTIP1 and DSNTIP2.

For data sharing installations, the work file database is the name you specified for WORK FILE DB on installation panel DSNTIPK. When you use the ENABLE or MEMBER functions, the steps to bind the DSNTIAD program are unnecessary and are edited out.

You must ensure that the installation jobs run on the MVS system where the appropriate DB2 subsystem is running. See page 2-212 for more information.

The DSNTIJTM job creates data sets in the work file database to be used as working storage for table spaces that require 4KB and 32KB buffering.

To create these data sets, the DSNTIJTM job uses as input the values you specified for the TEMP 4K SPACE option and the TEMP 32K SPACE option on the Sizes Panel (DSNTIPD).

To specify the number of 4KB and 32KB table spaces, the DSNTIJTM job uses as input the values you specified for TEMP 4K DATA SETS and TEMP 32K DATA SETS on the same panel. The naming convention for these data sets is provided by the installation CLIST:

```
vcatalog.DSNBC.DSNDDB07.DSNkkn.I0001.A001
```

where *vcatalog* is the catalog alias name you specified for the CATALOG ALIAS field on installation panel DSNTIPA2, *kk* is either 4K or 32K, and *nn* is the number of the data set. For example, if you specify 2 as the number of temporary 4KB data sets and 1 as the number of temporary 32KB data sets, DSNTIJTM defines the following table spaces:

```
vcatalog.DSNDBC.DSNDB07.DSN4K01.I0001.A001
vcatalog.DSNDBC.DSNDB07.DSN4K02.I0001.A001
vcatalog.DSNDBC.DSNDB07.DSN32K01.I0001.A001
```

If you do not need working storage for 32KB buffering, you can specify 0 for TEMP 32K SPACE and related options on installation panel DSNTIPD.

The TEMP 32K SPACE option determines the total size of all 32KB data sets. If you specify zero for this option, job DSNTIJTM does not contain a statement to create a data set for 32KB buffering. Be aware, however, that joins of smaller tables can produce rows longer than 4KB. For this reason, you might want 32KB data sets.

You can increase the number of additional temporary table spaces, increasing the values you specify for the TEMP 4K DATA SETS or TEMP 32K DATA SETS options, particularly if you expect a great deal of sorting at your site. Additional temporary table spaces can improve DB2 performance by reducing device contention among applications. These additional work files also can be used for sorting indexes on large tables when creating an index. For more information on adding temporary work files, see “DASD Requirements for the Work File Database” on page 2-21. You can choose to have job DSNTIJTM create these additional table spaces, or you can create them after you run the job.

To create the temporary table spaces after you run job DSNTIJTM, comment out all but steps DSNTTMP and DSNTIST from the job. Remove the COND statements, and change DSN4K01, DSN32K01, or both, to the table space names you want to use. Then, run the edited steps.

If job DSNTIJTM runs successfully, it produces the following return codes:

*Table 62. DSNTIJTM Return Codes*

| Step    | PROCSTEP | Return code  |
|---------|----------|--------------|
| DSNTIAD | PC       | 0000         |
|         | ASM      | 0000         |
|         | LKED     | 0000         |
| DSNTIAB |          | 0000         |
| DSNTIAS |          | 0000 or 0012 |
| DSNTICR |          | 0000         |
| DSNTTMP | DSNTIC   | 0000         |
| DSNTIST |          | 0000         |

The first time you run the DSNTIJTM job, you almost always get a return code of 12 for step DSNTIAS. This is because step DSNTIAS issues a command to stop DSNDB07 (the work file database), which has yet to be defined. A return code of 0 is issued if the job failed the first time and has to be rerun. When migrating, you also might receive a return code of 12 if the work file database DSNDB07 was dropped before running job DSNTIJTM.

Because this is the first use of DB2, errors from earlier steps may be detected here.

If you receive an abend reason code from the data manager (X'00C9'xxxx) or buffer manager (X'00C2'xxxx), carefully recheck jobs DSNTIJIN and DSNTIJID.

---

## Installation Step 15: Define and Bind DB2 Objects and User-Maintained Databases: DSNTIJSG

The default storage group is for user-defined DB2 tables that are not specifically assigned to a storage group. To create the default storage group and to bind DB2 plans for DCLGEN and SPUFI, run job DSNTIJSG.

Before you run this job, you can examine it and change the following items:

- The volume for the default storage group.
- The authorization level. You can make it more restrictive.
- Miscellaneous authorizations. You can add additional authorization to other buffer pools.

**If you use a product that uses a semicolon as a delimiter:** The CLIST adds SQL statements to job DSNTIJSG, but products that use a semicolon as a delimiting character cause semicolons to be removed from the installation CLIST before it is executed. To correct the problem, replace the semicolons at the end of each SQL statement in job DSNTIJSG before you run the job.

If you want to use the Data Facility Storage Management Subsystem to control the placement of data across volumes, edit job DSNTIJSG to replace the volume serial with '\*', as in the following statement:

```
CREATE STOGROUP SYSDEFLT VOLUMES ('*') ...
```

When you run the DSNTIJSG job, it does the following:

- Binds DB2 plans for SPUFI and DCLGEN.
- Creates the default storage group, which is used for your database, table space, and table definitions that are not related to a specific storage group.
- Grants use of the default buffer pool and storage group to PUBLIC.
- Grants use of the SYSDEFLT table space. This table space does not really exist, but this GRANT statement is necessary to give users the ability to implicitly create table spaces in the default database.
- Grants authority to create tables and table spaces in the default database to PUBLIC.
- Grants execute privilege on the plans for the SPUFI, DCLGEN, and DSNTIAD sample program to PUBLIC.
- Grants SELECT authority to the dummy table, SYSIBM.SYSDUMMY1 in the DSNDB06.SYSSTR table space.
- Defines the following DB2-supplied stored procedures:
  - # **DSNUTILS** Utility invocation stored procedure
  - # **DSNWZP** Subsystem parameter stored procedure

# See Appendix B of *Utility Guide and Reference* for information about stored  
# procedures that perform utility functions.

The DSNTIJSJG job also does the following for user maintained database activity:

- Creates the resource limit facility database. See Section 5 (Volume 2) of *Administration Guide* for more information.
- Creates the data definition control support database. For more information, see Section 3 (Volume 1) of *Administration Guide*.

The DSNTIJSJG job inserts a blank row into the communication database (CDB) table SYSIBM.LUNAME. The CDB holds tables containing information about your connection with remote DB2 subsystems. A blank row allows all SNA clients access to DDF. TCP/IP remote clients can not be controlled using the CDB. For more information about starting DDF, see “Distributed Data Facility Panel: DSNTIP5” on page 2-205. Field 3 of panel DSNTIP5, **TCP/IP ALREADY VERIFIED**, defines the minimum security requirements for all TCP/IP clients because inbound security requirements cannot be established on individual clients. For instructions on how to populate these tables, see “Step 4: Populate the Communications Database” on page 3-20 for VTAM information and “Step 4: Populate the Communications Database” on page 3-56 for TCP/IP information.

If the DSNTIJSJG job runs successfully, it produces return codes of 0. It can also produce a return code of 4 since a step within this job attempts to delete a row from a table that might not exist at the time this job is run. Expect the following messages from the BIND statement for each object provided by DB2:

```
DSNE932I WARNING, ONLY IBM-SUPPLIED PLAN NAMES SHOULD BEGIN WITH DSN
```

```
DSNE932I WARNING, ONLY IBM-SUPPLIED PACKAGE-IDS SHOULD BEGIN WITH DSN
```

```
DSNE932I WARNING, ONLY IBM-SUPPLIED COLLECTION-IDS SHOULD BEGIN WITH DSN
```

If the DSNTIJSJG job fails or abends, be sure that the user specified on the JOB statement is authorized. Use the same name you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP.

Correct any other problems with the DSNTIJSJG job, uncomment the rerun step, and rerun it. If there are not enough resources to run the job, review the values you specified for the DB2 installation parameters (see job DSNTIJUZ). Use the standard update procedure to make any necessary modifications. Refer to “The Update Process” on page 2-228 for information on the standard update procedure. Then stop DB2, rerun the DSNTIJUZ job, start DB2, and rerun the DSNTIJSJG job.

---

## Installation Step 16: Populate the User-Maintained Databases (Optional)

DSNTIJSJG creates user-maintained databases that need to be populated. These include the resource limit specification table, and the data definition control support tables. See the following sections for more information:

- Section 5 (Volume 2) of *Administration Guide* for information on the resource limit specification table
- Section 3 (Volume 1) of *Administration Guide* for information on the data definition control support tables.

---

## # Installation Step 17: Bind the Packages for DB2 REXX Language Support: DSNTIJRX

- # Before you can use DB2 REXX Language Support, you must bind DB2 packages that DB2 REXX Language Support uses. Run job DSNTIJRX to do this. Before you run DSNTIJRX, make the following changes:
- # • Add a job statement.
  - # • Change DSN SYSTEM(DSN) to DSN SYSTEM(ssid), where ssid is the name of the DB2 subsystem on which you will use DB2 REXX Language Support.
  - # • Change all instances of DSN!!0 to your DB2 data set name prefix.
  - # • Change all instances of DSNTIA!! to the plan name for the DSNTIAD program.

---

## Installation Step 18: Back Up the DB2 Directory and Catalog: DSNTIJIC

### Attention:

You need to create a backup copy of the DB2 directory and catalog. DB2 starts if you do not have backup copies, but if errors occur in the directory or catalog that require you to reinstall DB2, you lose all your tables and data. We recommend that you copy the catalog and directory at least daily if you make any changes in them. Recovery for these databases is longer if the copies are not current, and the entire subsystem is affected.

For data sharing considerations, see *Data Sharing: Planning and Administration*.

To create a copy of the DB2 directory and catalog, run the DSNTIJIC job. Before you run the DSNTIJIC job, examine the job for the following:

- The tape unit name. The job lists the tape unit name as TAPE. If this is incorrect for your site, correct it. The name TAPE is also the unit name for the default archive log data sets.
- Expiration date or retention period. You can add a retention period or an expiration date to the job.
- The user on the JOB statement. Make sure the user is authorized. This must be the same user you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP.

The DSNTIJIC job contains a list of all the DB2 directory and catalog table spaces. When you run job DSNTIJIC, it invokes the DB2 image copy utility to copy these table spaces to tape. The copied table spaces allow you to recover the DB2 catalog and DB2 directory in case of a failure.

If the DSNTIJIC job fails or abends, verify that there are no problems with the tape setup for image copy. If this does not fix the problem, examine the utility job output (JOBLOG) or the console log for problems. For instance, make sure there are no incorrect sizes or I/O errors.

Run the DSNTIJIC job periodically—perhaps daily or weekly. This reduces the amount of time required for recovering the directory or catalog. The copied data and log data sets are needed for recovery.



## Authorizing DB2 Users

After you complete all the steps above, DB2 is available to TSO.

During the ISPF tailoring session, you named one or two IDs to have installation SYSADM authority. One of these users can now grant various levels of authority to other users. You can use SPUFI or a job similar to DSNTIJSG (described on page 2-254) to perform the authorization. For suggestions about using DB2 authorities and privileges, see Section 3 (Volume 1) of *Administration Guide*.

---

## Installation Step 19: Verify the Installation

Run the sample applications to verify that your installation process has been successful. Select the phases you need to run based on the attachment facilities you installed, the languages you use, and whether the sample objects exist. For more information, see “Chapter 2-9. Verifying with the Sample Applications” on page 2-305.

---

## Installing a Communications Network

If you plan to use the distributed data facility (DDF), we recommend you become familiar with the DDF function and install Virtual Telecommunications Access Method (VTAM) (required for DDF), or OpenEdition TCP/IP support before loading the data sets into DB2 libraries. For information on installing VTAM, see “Chapter 3-2. Connecting Systems with VTAM” on page 3-9. For information on installing OpenEdition TCP/IP support, see “Chapter 3-3. Connecting Systems with TCP/IP” on page 3-49.

The first step in accessing distributed data is to install VTAM if it is not installed already. The minimum VTAM release that DDF supports is Version 3 Release 3 in either MVS environment. For information about planning your network configuration, see *Planning for NetView, NCP, and VTAM*. For information about installing your VTAM system, see *VTAM for MVS/ESA Network Implementation Guide*. For factors that need to be considered before installing or customizing VTAM, see Section 4 (Volume 1) of *Administration Guide*.

For information about customizing VTAM to work optimally with DB2, see “Step 1: Install VTAM Version 3.4.2 or Version 4.1” on page 3-11.

**Installing NetView:** If you use DDF, you might want to install NetView so that DB2 can send alerts to NetView if DB2 detects security exposures or protocol errors. For information about installing NetView, see *NetView Installation and Administration Guide*.

---

## Installing a Second DB2 on the Same Operating System

This section tells how to install a second DB2 subsystem on an operating system. There are two major sections:

- Planning considerations
- Installing procedure

## Planning Considerations

The primary consideration in planning for a second DB2 subsystem is its purpose. Using a second subsystem has a substantial impact on your environment. Second DB2 subsystems are not uncommon; they have been used to:

- Run separate service levels of the code. This can provide more extensive testing of preventive service before use with a production system.
- Run separate releases of the code. However, two different releases of DB2 on the same system must have separate libraries.
- Separate test and production activities. This setup can improve DB2's performance and availability for production.

For example, suppose the processor that runs your production subsystem fails. If your test subsystem is on another processor, you can stop the test subsystem and start the production subsystem on that processor. This only works if the requisite DB2 data sets are on shared DASD and you have used global resource serialization (or an equivalent) to protect the production DB2 data sets.

- Prevent access by one class of users to certain data. If this is your primary purpose, it is probably worthwhile to reconsider the DB2 authorization scheme.

Table 63 lists other considerations in planning for a second DB2 subsystem.

Table 63. Considerations for Installing a Second DB2Subsystem

| Considerations                         | Decisions You Need to Make                                                                                         | Where to Find Information |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------|---------------------------|
| Libraries                              | A single, shared library or separate libraries                                                                     | Page 2-259                |
| RACF protection                        | Resource and ID for the two subsystems                                                                             | Note <sup>1</sup>         |
| DB2 logging                            | BSDS, active, and archive log space requirements                                                                   | Page 2-18                 |
| Database space requirements            | DB2 directory, DB2 catalog, and user data                                                                          | Page 2-16                 |
| Performance                            | Processor and main storage use                                                                                     | Note <sup>2</sup>         |
| Distributed data facility requirements | Coordination of location names, logical unit names, and network passwords with remote DB2 subsystems               | Page 3-3                  |
| Common service area (CSA) requirement  | DB2 and the IRLM                                                                                                   | Page 2-35                 |
| Shared DASD                            | Giving different names to active and archive log data sets, or including those data sets in the GRS inclusion list | Page 2-259                |

**Note:**

<sup>1</sup> See Section 3 (Volume 1) of *Administration Guide*

<sup>2</sup> See Section 5 (Volume 2) of *Administration Guide*

Each subsystem must have a separate *prefix.SDSNEXIT* library. Sharing the *prefix.SDSNSAMP* library requires coordination to avoid overlaying parameter members.

In an environment where DB2 uses shared DASD, use different data set names for the active and archive log data sets if possible. If this is not possible, include those data sets in the global resource serialization (GRS) inclusion list. See “Considerations for Using Shared DASD” on page 2-263 for more details.

## Installation Procedure

This section describes the principal steps to take when installing a second DB2 subsystem.

### Loading DB2 Libraries

This step is required if you want separate libraries for the two systems. If the systems have different code releases or different service levels, they must have separate libraries. You must plan the space for each library separately and load the libraries, using different prefixes for each library and for the SMP/E data sets or separate SMP/E zones.

### Tailoring Installation Jobs

You can use the procedure described in “Chapter 2-5. Installing, Migrating, and Updating System Parameters” on page 2-89 to specify appropriate parameter values for the second subsystem. Table 64 shows the parameter values you can change and the parameter values you must change during the installation process. The figure includes the panel on which the parameter appears, the panel parameter name, and any comments that pertain to the parameter.

Table 64 (Page 1 of 2). Parameters to Change When Installing the Second Subsystem

| Installation Panel | Action                                                                                |
|--------------------|---------------------------------------------------------------------------------------|
| DSNTIPA0           | Check all values and make appropriate changes.                                        |
| DSNTIPA1           | For separate libraries, change LIBRARY DATA SET NAME PREFIX and DATA SET NAME PREFIX. |
| DSNTIPA2           | You must change CATALOG ALIAS. You can change VOLUME SERIALS 1 through 6.             |
| DSNTIPK            | Check all values and make appropriate changes.                                        |
| DSNTIPH            | Check all values and make appropriate changes.                                        |
| DSNTIPT            | Check all values and make appropriate changes.                                        |
| DSNTIPU            | Check all values and make appropriate changes.                                        |
| DSNTIPQ            | Check all values and make appropriate changes.                                        |
| DSNTIPG            | Check all values and make appropriate changes.                                        |
| DSNTIPW            | Check all values and make appropriate changes.                                        |
| DSNTIPV            | Check all values and make appropriate changes.                                        |
| DSNTIP3            | Check all values and make appropriate changes.                                        |
| DSNTIPD            | Check all values and make appropriate changes.                                        |
| DSNTIPE            | Check all values and make appropriate changes.                                        |
| DSNTIP1            | Check all values and make appropriate changes.                                        |
| DSNTIP2            | Check all values and make appropriate changes.                                        |
| DSNTIPN            | Check all values and make appropriate changes.                                        |
| DSNTIPO            | Check all values and make appropriate changes.                                        |
| DSNTIPF            | Check all values and make appropriate changes.                                        |

Table 64 (Page 2 of 2). Parameters to Change When Installing the Second Subsystem

| Installation Panel | Action                                                                                                                 |
|--------------------|------------------------------------------------------------------------------------------------------------------------|
| DSNTIP4            | Check all values and make appropriate changes.                                                                         |
| DSNTIPI            | You must change SUBSYSTEM NAME.                                                                                        |
| DSNTIPJ            | Check all values and make appropriate changes.                                                                         |
| DSNTIPP            | Change integrated catalog facility catalog to match the new integrated catalog facility catalog. Change all passwords. |
| DSNTIPM            | Change SUBSYSTEM NAME and SUBSYSTEM PREFIX.                                                                            |
| DSNTIPL            | Check all values and make appropriate changes. You must change data set names and prefixes.                            |
| DSNTIPA            | Check all values and make appropriate changes. You must change data set names and prefixes.                            |
| DSNTIPS            | Check all values and make appropriate changes. You must change data set names and prefixes.                            |
| DSNTIPR            | Check all values and make appropriate changes. You must change names and password.                                     |
| DSNTIP5            | Check all values and make appropriate changes.                                                                         |
| DSNTIPX            | Check all values and make appropriate changes.                                                                         |
| DSNTIPZ            | Check all values and make appropriate changes.                                                                         |
| DSNTIPY            | Check all values and make appropriate changes.                                                                         |
| DSNTIPC            | Check EDMPOOL value; no other changes can be made.                                                                     |

## Installing a Second DB2

The following list shows the jobs that must be run, or might need to be run, when installing the second DB2 subsystem.

| Job      | Comments                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DSNTIJCA | Run this job if you are defining a new integrated catalog facility catalog.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| DSNTIJIN | Run this job. It allocates the data sets listed below. These data sets contain the sample plans and programs.<br><br><i>prefix</i> .DBRMLIB.DATA<br><i>prefix</i> .SRCLIB.DATA<br><i>prefix</i> .RUNLIB.LOAD<br><br>Two subsystems cannot share these data sets. If there is undetected sharing, such as both subsystems using the same log, then data could be lost. If you use the same library prefix for both subsystems, then change the name of the data sets, unless you do not need them on the first subsystem. Later jobs overwrite them, and then the plans bound in the first subsystem will not work with the new load modules. |
| DSNTIJID | Run this job to initialize the DB2 data sets.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| DSNTIJMV | Add another subsystem name and subsystem recognition character to IEFSSNxx. LNKLSTxx modifications are needed only for separate libraries.<br><br>For separate libraries, add STEPLIB statements to the precompile and bind steps for program preparation. Add STEPLIB statements for                                                                                                                                                                                                                                                                                                                                                        |

the DB2 offline utilities. Choose a naming convention for any new procedures, and change those as needed.

For a single library, you must add the exit module data set (*prefix.SDSNEXIT*) to the STEPLIB statement to contain your changed subsystem parameter. Put this data set first in the STEPLIB concatenation.

**DSNTIJUZ** Run this job, without changes if you use separate libraries. For a single library, provide a separate *prefix.SDSNEXIT* data set for each subsystem.

The default SSID displayed by certain panels and procedures is the same for every subsystem. Make sure that the correct subsystem is specified in these cases.

If you are using Resource Access Control Facility (RACF), you can define new user profiles and IDs to provide a separate level of security for each DB2 subsystem. See Section 3 (Volume 1) of *Administration Guide* for information.

## Connecting Attachment Facilities

This section contains information about connecting the attachment facilities for the second DB2 subsystem.

**Connecting the TSO Attachment Facility:** The following list shows the procedures to follow when connecting the TSO attachment facility.

| Procedure            | Comments                                                                                                                                                                                                                                                                                             |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Logon procedure      | Add STEPLIB statements if you have decided to have a separate library for the second subsystem or manage two different SDSNEXIT data sets. The STEPLIB statement must be authorized using the authorized program facility (APF).<br><br>A DB2 logon procedure can use only one set of DB2 libraries. |
| Concatenate CLISTs   | Update the SYSPROC library if you decide to have a separate library for the second subsystem.                                                                                                                                                                                                        |
| Concatenate panels   | Update the ISPPLIB library if you decide to have a separate library for the second subsystem.                                                                                                                                                                                                        |
| Concatenate messages | Update the ISPMLIB library if you decide to have a separate library for a second subsystem.                                                                                                                                                                                                          |
| Customize panel      | You must update the ISPF primary option panel (ISR@PRIM) if you decide to have a separate library for the second subsystem.                                                                                                                                                                          |
| Authorize users      | Grant authorization on both subsystems as necessary.                                                                                                                                                                                                                                                 |
| DB2I defaults panel  | Specify the new subsystem name as required.                                                                                                                                                                                                                                                          |

**Connecting the IMS Attachment Facility:** The following list shows the procedures to follow when connecting a second IMS attachment facility.

| <b>Procedure</b>   | <b>Comments</b>                                                                                       |
|--------------------|-------------------------------------------------------------------------------------------------------|
| STEPLIB            | For separate libraries, change this DD statement to refer to the new libraries.                       |
| DFSESL             | For separate libraries, change this DD statement to refer to the new libraries.                       |
| Subsystem member   | Define a new subsystem name, language interface token (LIT), and command recognition character (CRC). |
| Language interface | Define a new LIT and reassemble.                                                                      |
| Linkage editor JCL | Specify the library containing the new language interface.                                            |
| Authorize users    | Perform this procedure.                                                                               |

### **Preparing DB2 for Use**

The following list explains how to prepare the subsystem for use when two DB2 subsystems were installed.

| <b>Procedure</b> | <b>Comments</b>                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IPL MVS          | This step is required to make any SYS1.PARMLIB changes take place.                                                                                                                                                                                                                                                                                                                                                             |
| Start DB2        | Use the new command prefix (formerly called the subsystem recognition character) that you named during the installation process.                                                                                                                                                                                                                                                                                               |
| DSNTIJIC         | Run this job.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| DSNTIJTM         | Run this job.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| DSNTIJSG         | If you use the same library prefix for both subsystems, then change the name of the data sets listed below, unless you do not need them on the first subsystem. Later jobs write information in them and prevent use of the new load module DSNTIAD with the previously bound plan DSNTPE51 on the first subsystem.<br><br><i>prefix.DBRMLIB.DATA</i><br><i>prefix.RUNLIB.LOAD</i>                                             |
| DSNTEJxx         | If you use the same library prefix for both subsystems, then change the name of the data sets listed below, unless you do not need them on the first subsystem. Later jobs overwrite information in them and prevent use of the new load modules of sample programs with the previously bound sample plans on the first subsystem.<br><br><i>prefix.DBRMLIB.DATA</i><br><i>prefix.SRCLIB.DATA</i><br><i>prefix.RUNLIB.LOAD</i> |

### **Verifying Your Installation Process**

Run the sample applications to verify that your installation process has been successful. Select the phases you need to run based on the attachment facilities you installed, the languages you use, and whether the sample objects exist. For more information, see "Chapter 2-9. Verifying with the Sample Applications" on page 2-305.

---

## Considerations for Using Shared DASD

These considerations apply to non-data sharing environments only. If you use a data sharing environment, the logs must be on shared DASD. If you plan to share DASD among DB2 subsystems, avoid problems with your active and archive log data sets by taking the following precautions:

- Make sure each subsystem has unique log data set names. This prevents situations like the following:

Subsystem A on operating system 1 and subsystem B on operating system 2 share the same MVS catalog name, and their log data set names are the same. You start subsystem B while subsystem A is still running on operating system 1. This causes log data sets to be allocated for subsystem A even though they already exist.

- Use global resource serialization (GRS), or an equivalent, and include your active and archive log data sets in the GRS inclusion list. This prevents situations like the following:

Subsystem A on operating system 1 and subsystem B on operating system 2 share DASD, and the active log is in a shared DASD volume. Subsystem B fails. You attempt to start subsystem B, but you accidentally start subsystem A on operating system 2 even though it is still running on operating system 1. This causes log data sets to be allocated for subsystem A even though they already exist.

---

## Enabling stored procedures after installation

To enable stored procedures *after* you have completed the installation process choose one of the following methods:

### 1. Run the installation CLIST in INSTALL or MIGRATE mode

- On panel DSNTIPA1, specify the input member that contains field values for your current installation.
- In the remaining installation panels, leave the existing values, except in the following cases:
  - On panel DSNTIPT, choose a different name for TEMP CLIST LIBRARY and SAMPLE LIBRARY to avoid overwriting your original libraries.
  - On panel DSNTIPG, specify
    - a data set name for LE/370 RUN TIME LIBRARY
    - data set names for PL/I for MVS & VM (you need PL/I for MVS & VM to run the PL/I stored procedures sample applications)
  - On panel DSNTIPX, fill in all fields for the type of stored procedure environment you will use.
  - On panel DSNTIPY, specify a remote location name. (This name is used for the stored procedure sample applications).
  - When the installation CLIST completes, obtain the updated copy of *ssnmSPAS* from job DSNTIJMV and install it in your PROCLIB.
    - For DB2-established stored procedures:

- Run the new copy of DSNTIJUZ.
  - Restart DB2 with the new parameters.
  - Authorize the ID associated with the stored procedure startup procedure to use CAF.
- For WLM-established stored procedures:
- Run the new copy of DSNTIJUZ.
  - Restart DB2 with the new parameters.
  - Define JCL procedures for the stored procedures address spaces
- Member DSNTIJMV of data set DSN510.SDSNSAMP contains sample JCL procedures for starting WLM-established address spaces.
- Define WLM application environments for groups of stored procedures and associate a JCL startup procedure with each application environment.
- See Section 5 (Volume 2) of *Administration Guide* for information on how to do this.

## 2. Edit DSNTIJUZ

- Edit job DSNTIJUZ to add or change values of the stored procedure parameters: STORMXAB, STORPROC, and STORTIME
- Run DSNTIJUZ.
- Restart DB2 with the new parameters.
- For DB2-established stored procedures,
  - generate a *ssnm*SPAS PROC supplying the pertinent information: PROCNAME, SUBSYS name, NUMTCB, STEPLIB libraries. Use procedure DSNPAS in DSN510.SDSNSAMP as a model.
  - Authorize the ID associated with the stored procedure to use CAF.
- For WLM-established stored procedures
  - Define JCL procedures for the stored procedures address spaces

Member DSNTIJMV of data set DSN510.SDSNSAMP contains sample JCL procedures for starting WLM-established address spaces.

  - Define WLM application environments for groups of stored procedures and associate a JCL startup procedure with each application environment.

See Section 5 (Volume 2) of *Administration Guide* for information on how to do this.

The second method does not give you a migration path since your DSNTIDxx member and DSNTIJUZ parameters are not saved for future input. In addition, this second method does not generate the sample jobs DSNTEJ6S, DSNTEJ6P, DSNTEJ6D, DSNTEJ6T, DSNTEJ61, and DSNTEJ62 since you are not running the DSNTINST CLIST.

For more information on stored procedures see Section 6 of *Application Programming and SQL Guide* .



## # **Enabling DB2-supplied stored procedures after installation**

# You can enable the DB2-supplied stored procedures after you have completed the  
# installation process. The DB2-supplied stored procedures are:

# **DSNACCQC** DB2 UDB Control Center catalog query stored procedure

# **DSNACCAV** DB2 UDB Control Center partition information stored procedure

# **DSNUTILS** Utility invocation stored procedure

# **DSNWZP** Subsystem parameter stored procedure

# If you have already installed the DB2 UDB Control Center FMID, JDB551D , use  
# job DSNTIJCC to define the DSNACCQC and DSNACCAV stored procedures.  
# Because DSNTIJCC is not tailored by the installation process, you need to make  
# the following changes before you run DSNTIJCC:

- # • Add a JOB statement.
- # • Change all instances of DSN!!0.SDSNLOAD to the name of your DB2 load  
# library.
- # • Change all instances of DSN!!0.RUNLIB.LOAD to the name of your application  
# load library.
- # • Change all instances of DSN!!0.SDSNDBRM to the name of your DB2 DBRM  
# library.
- # • Change all instances of DSNTIA!! to the plan name for program DSNTIAD.
- # • In all instances of SYSTEM(DSN), change DSN to your DB2 subsystem name.

# For more information on stored procedures see Section 6 of *Application*  
# *Programming and SQL Guide*.



---

## Chapter 2-7. Migrating the DB2 Subsystem

This chapter describes the steps necessary to migrate from DB2 for MVS/ESA Version 4 to DB2 for OS/390 Version 5. DB2 for MVS/ESA Version 4 is the only release from which you can migrate to DB2 for OS/390 Version 5.

Before you begin migrating, you need to load the DB2 libraries. This process is described beginning on page 2-69. You also need to run the installation CLIST. The process is described beginning on page 2-89. Also see “Migration Considerations” for information on changes that might affect your migration process.

When you migrate to Version 5, avoid using new DB2 facilities until you are not likely to fall back.

### Attention:

If documented procedures are not followed or if you try to migrate from a release other than DB2 for MVS/ESA Version 4, unpredictable results can occur after migration.

Make sure that your Version 4 subsystem is at the proper service level. Refer to *IBM DATABASE 2 Program Directory* shipped with the product for keyword specifications for preventive service planning (PSP). Check Information/Access or the ServiceLink facility of IBMLink for PSP information before you migrate and monthly for access to the most current information about DB2.

You must not use secondary authorization IDs to perform any of the following migration steps.

---

## Migration Considerations

Be aware of the following changes that might affect your migration to Version 5. Your flagger, schema processor and DCLGEN plans from Version 4 are usable in Version 5 only if they were bound as packages or if PTF UN98205 was applied to Version 4. If you have Version 4 plans that require conversion to packages, do the conversion **before** you begin the migration process. For information on how migration might affect your DB2 operations, see “Migration Step 24: Make Adjustments for Release Incompatibilities” on page 2-290.

## WLM-established Stored Procedures Address Space

New columns are added to SYSIBM.SYSPROCEDURES to support WLM-established stored procedures address spaces:

- COMMIT\_ON\_RETURN
- EXTERNAL\_SECURITY
- PGM\_TYPE
- WLM\_ENV

These steps migrate your existing stored procedures to a WLM-established environment:

1. Relink all of your stored procedures with the OS/390 RRS application language interface.
2. Use WLM to define WLM application environments.

3. Update SYSIBM.SYSPROCEDURES with the WLM\_ENV name.
4. Issue STOP PROC followed by START PROC.

For more information, refer to *Application Programming and SQL Guide*.

## GRANTs on SYSCAT.PROCEDURES Lost

Because of internal changes to the view, any GRANTs on this catalog view are lost after migration. Issue this query to determine which GRANTs you need to reissue:

```
#
SELECT * FROM SYSIBM.SYSTABAUTH
WHERE TCREATOR='SYSCAT' AND TTNAME='PROCEDURES';
```

## Selective Partition Locking

You can limit partition locking to improve performance even if you have not enabled data sharing. All plans bound with ACQUIRE(USE) plus packages and SQL that access partitioned table spaces automatically begin using this feature.

Selective partition locking is not supported for these cases:

- A type 1 index is accessed
- Plans bound with ACQUIRE(ALLOCATE)
- Table space defined with LOCKSIZE TABLESPACE
- LOCK TABLE IN EXCLUSIVE MODE is used
- CREATE, ALTER, or DROP cause the table space to be locked

## Work File Database Size Calculations

The migration job DSNTIJTC creates and updates indexes on catalog tables. These indexes are created **sequentially** during migration. The work file database is used for the sort. The migration fails if you do not have enough storage, so ensure that you have enough space before you begin. See “DASD Requirements for the Work File Database” on page 2-21 for information about space requirements.

Table 65 shows the indexes that are new and changed.

Table 65. Indexes added or updated sequentially using the work file database

| Index name      | Catalog table name | Column names                                                                                                                                                                |
|-----------------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SYSIBM.DSNFPX01 | SYSIBM.IPNAMES     | LINKNAME                                                                                                                                                                    |
| SYSIBM.DSNFCX01 | SYSIBM.LOCATIONS   | LOCATION                                                                                                                                                                    |
| SYSIBM.DSNFNX01 | SYSIBM.LUNAMES     | LUNAME                                                                                                                                                                      |
| SYSIBM.DSNFEX01 | SYSIBM.USERNAMES   | TYPE, AUTHID (descending), LINKNAME (descending)                                                                                                                            |
| SYSIBM.DSNFLX01 | SYSIBM.LULIST      | LINKNAME, LUNAME                                                                                                                                                            |
| SYSIBM.DSNFLX02 | SYSIBM.LULIST      | LUNAME                                                                                                                                                                      |
| SYSIBM.DSNFMX01 | SYSIBM.LUMODES     | LUNAME, MODENAME                                                                                                                                                            |
| SYSIBM.DSNFDX01 | SYSIBM.MODESELECT  | LUNAME, AUTHID (descending), PLANNAME (descending)                                                                                                                          |
| SYSIBM.DSNATX02 | SYSIBM.SYSTABAUTH  | GRANTEE, TCREATOR, TTNAME, GRANTEETYPE, UPDATECOLS, ALTERAUTH, DELETEAUTH, INDEXAUTH, INSERTAUTH, SELECTAUTH, UPDATEAUTH, CAPTUREAUTH, REFERENCESAUTH, REFCOLS <sup>1</sup> |

**Note:** <sup>1</sup> DSNATX02 is updated with column REFCOLS during migration.

## REPORT RECOVERY Changes

REPORT RECOVERY with the CURRENT option now shows only SYSLGRNX rows and archive log volumes created since the last incremental image copy.

REPORT RECOVERY without the CURRENT option now marks archive log volumes created after the last FULL IMAGE COPY, LOAD REPLACE LOG YES, or REORG LOG YES with an asterisk, giving you the option to prepare additional log tapes beyond what is needed for recovery.

## # RUNSTATS Changes

# After applying APAR PQ21014, RUNSTATS generates uniform and non-uniform  
# distribution statistics in SYSCOLDIST. Prior to APAR PQ21014 only non-uniform  
# statistics were stored in SYSCOLDIST. After applying APAR PQ21014, RUNSTATS  
# may generate more rows in SYSCOLDIST.

## Entry SQL Conformance

Migration to several SQL enhancements requires special consideration.

- Semantics of the WITH CHECK OPTION clause has different meaning in Version 5 than in previous releases when it was the only valid syntax for defining a check option for a view on the CREATE VIEW statement.

In Version 5, WITH CHECK OPTION has equivalent meaning to WITH CASCADED CHECK OPTION (new default). The meaning of WITH CHECK OPTION in previous release is equivalent to WITH LOCAL CHECK OPTION in Version 5.

Therefore, views from a previous release defined with WITH CHECK OPTION that are dropped and re-created with the same syntax in Version 5 will behave as if the WITH CASCADED CHECK OPTION is specified. Recreating a view with the same semantics as in previous releases requires the LOCAL keyword to be explicitly specified.

To identify views that might be a problem if re-created in Version 5, see “Dynamic SQL Statements (DSNTESA, DSNTESQ)” on page 2-363.

- When the standards option is in effect, the SELECT privilege is required on the table being updated or deleted from, in addition to the UPDATE or DELETE privilege.
- When the Schema Processor processes a CREATE TABLE statement with either a PRIMARY KEY or UNIQUE clause, DB2 creates an index using default specifications.
- Plans and packages are no longer invalidated when ALTER TABLE is used to add or drop a primary key.
- In certain error situations, an SQLSTATE value returned to an application differs from the value returned when the same SQL statement is executed using a prior release of DB2. See *SQL Reference* for details.
- In certain error situations when the standards option is in effect, the SQLCODE value returned to an application will differ from the value returned when the same SQL statement is executed using a prior release of DB2. See *SQL Reference* for details.

## Catalog Column Changes

The integer columns of catalog tables that have updatable statistics columns are changed to floating point columns. Columns that violate the new check constraints are set to -1. You might want to execute queries similar to the following to see what is changed during migration:

General-use Programming Interface

```
SELECT TBNAME, FREQUENCY, FREQUENCYF, NAME, CARDF, NUMCOLUMNS
FROM SYSIBM.SYSCOLDIST
WHERE CARDF < -1 OR NUMCOLUMNS < 1 OR
 FREQUENCY < 0 OR FREQUENCY > 10000 OR
 FREQUENCYF < 0 OR FREQUENCYF > 1;

SELECT * FROM SYSIBM.SYSCOLDISTSTATS
WHERE CARDF < -1 OR NUMCOLUMNS < 1 OR
 FREQUENCY < 0 OR FREQUENCY > 10000 OR
 FREQUENCYF < 0 OR FREQUENCYF > 1 OR
 PARTITION < -1 OR PARTITION = 0;
```

End of General-use Programming Interface

There are new columns added to catalog tables SYSCOLDIST and SYSCOLDISTSTATS. These are TYPE, CARDF, NUMCOLUMNS and COLGROUPOCOLNO. There are also changes to several columns in both SYSCOLDIST and SYSCOLDISTSTATS. These columns contain statistics that are collected on concatenated key columns of an index. Applications that query, insert, update or delete from these catalog tables might need some modifications to work properly.

See Appendix D of *SQL Reference* for detailed catalog changes.

## Migrating a Member of a Data Sharing Group

Migrating a data sharing group requires a carefully thought out plan:

1. Read the information about migration considerations in this book and also in Chapter 4 of *Data Sharing: Planning and Administration*.
2. Make a plan to migrate the data sharing group over as short a time as possible.
3. Apply the fallback SPE to the DB2 for MVS/ESA Version 4 load library before attempting to migrate any member of the group.
4. Stop and restart each member to pick up the change.
5. Follow the procedure in "Migrating a Member of a Data Sharing Group." To prepare for fallback, keep the subsystem parameter load module used by DB2 for MVS/ESA Version 4.

These jobs are edited only for the first member of a data sharing group that migrates:

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| DSNTIJIC | DSNTIJVC | DSNTIJSU | DSNTIJIN | DSNTIJEX |
| DSNTIJTM | DSNTIJSG | DSNTIJTC | DSNTIJXC |          |

## Release Coexistence

This section highlights considerations for coexistence between Version 5 and previous releases in a data sharing and a distributed environment.

### Data Sharing

DB2 can support both DB2 for MVS/ESA Version 4 and DB2 for OS/390 Version 5 members in a data sharing group. To support both releases, you must apply the fallback SPE to all members of the group as described in “Migrating a Member of a Data Sharing Group” on page 2-270.

For the best availability, you can apply the SPE to one member at a time, and you can migrate the members to the new release one member at a time. When developing your migration plan, keep in mind that new functions introduced in this release are not available to members of the group that have not yet migrated. Thus, it is best to either:

- Migrate all members to the new release before you use new function.
- Restrict the execution of packages and plans bound on Version 5 to members of the group that have already migrated.

This restriction serves two purposes. First, if new plans or packages use new functions, you avoid the application errors that can occur if the plan or package tries to execute an SQL statement that is not allowed in DB2 for MVS/ESA Version 4. Secondly, you avoid the automatic rebind that occurs when any plan or package that is bound on Version 5 is run on Version 4.

You can attach to either release of DB2 with your existing TSO or CAF logon procedures, without changing the load libraries for your applications. After you migrate completely to the latest level of DB2, you **MUST** update those procedures and jobs to point to the latest level of DB2 load libraries.

### Distributed Environment

DB2 for OS/390 communicates in a distributed data environment with DB2 Version 2 Release 3 and later, using either DB2 private protocol access or DRDA access. However, the distributed functions introduced in this release of DB2 for OS/390 can be used only when using DRDA access.

Other DRDA partners can also take advantage of the functions introduced in this release of DB2 for OS/390.

### Sysplex Query Parallelism

To take advantage of Sysplex query parallelism, you must rebind plans or packages. We recommend not rebinding applications until all of the members of the data sharing group have migrated to DB2 for Version 5.

Some access paths might change when bound on Version 5 because DB2 does not pick a hybrid join in which SORTN\_JOIN=Y when DEGREE=ANY. This is to avoid automatic rebinds on members still on Version 4. Dynamic queries can use Sysplex query parallelism, assuming you have performed the tasks described in *Data Sharing: Planning and Administration*.

---

## # Migration Step 1: Run Link Checker on DB2 for MVS/ESA Version 4

### # Table Spaces (optional)

DSN1CHKR (link checker) is the utility that verifies the integrity of the DB2 directory and catalog table spaces. DSN1CHKR scans the specified table space for broken links, hash chains, and orphans (records that are not part of any link or chain). You need only run DSN1CHKR on catalog and directory table spaces that contain links or hashes. You must issue the STOP DATABASE command on these table spaces before running DSN1CHKR on them. These table spaces include:

- DSNDB06.SYSDBASE
- DSNDB06.SYSDBAUT
- DSNDB06.SYSGROUP
- DSNDB06.SYSPLAN
- DSNDB06.SYSVIEWS
- DSNDB01.DBD01

# In addition, we recommend that you run DSN1COPY with the CHECK option on all  
# of the catalog table spaces. This ensures that the table space pages are physically  
# correct and the catalog table spaces are clustered. When you run this utility on  
# segmented table spaces you might get message DSN1985I. The segmented table  
# spaces in the catalog and directory are: DSNDB06.SYSPACKAGE,  
# DSNDB06.SYSSTR, DSNDB06.SYSSTATS, DSNDB06.SYSDDF,  
# DSNDB01.SYSUTILX, and DSNDB01.SPT01. You can usually ignore this  
# message, see description of DSN1985I in *Messages and Codes* for details. We  
highly recommend running DSN1COPY and DSN1CHKR with the catalog and the  
directory table spaces stopped, or with DB2 stopped. We also recommend running  
the CHECK INDEX utility. For more information on DSN1CHKR and DSN1COPY,  
see Section 3 of *Utility Guide and Reference*. For more information on CHECK  
INDEX, see Section 2 of *Utility Guide and Reference*.

You should run the following query on your Version 4 catalog tables to ensure that you do not have a STOGROUP defined with both specific and non-specific volume IDs. If the query returns any rows, the STOGROUPs identified have both specific and non-specific volume IDs. Table spaces in databases that use these STOGROUPs can not be image copied or recovered until ALTER STOGROUP is used to remove volumes so the STOGROUP has either specific or non-specific volume IDs.

This query is commented out in member DSNTESTQ of *prefix.SDSNSAMP*.

General-use Programming Interface

```
SELECT * FROM SYSIBM.SYSVOLUMES V1
 WHERE VOLID ^= '*' AND
 EXISTS (SELECT * FROM SYSIBM.SYSVOLUMES V2
 WHERE V1.SGNAME = V2.SGNAME AND
 V2.VOLID = '*')
```

End of General-use Programming Interface



## Migration Step 2: Determine Which Plans and Packages Are Invalid after Migration (Optional)

Migrating to DB2 for OS/390 Version 5 renders some plans and packages invalid. Find out which ones are invalid by running the following queries on your DB2 for MVS/ESA Version 4 subsystem:

### General-use Programming Interface

```
SELECT DISTINCT DNAME
FROM SYSIBM.SYSPLANDEP
WHERE BNAME IN ('SYSCOLAUTH',
'SYSDATABASE', 'DSNDDH01', 'DSNDDX02',
'SYSDBAUTH', 'DSNADH01', 'DSNADX01',
'SYSDBRM',
'SYSINDEXES', 'DSNDXX01', 'DSNDXX02', 'DSNDXX03',
'SYSPLAN', 'DSNPPH01',
'SYSPLANAUTH', 'DSNAPH01', 'DSNAPX01',
'SYSRESAUTH', 'DSNAGH01', 'DSNAGX01',
'SYSSTOGROUP', 'DSNSSH01',
'SYSSYNONYMS', 'DSNDYX01',
'SYSTABAUTH', 'DSNATX01', 'DSNATX02', 'DSNATX03',
'SYSTABLESPACE', 'DSNDSX01',
'SYSUSERAUTH', 'DSNAUH01', 'DSNAUX02')
AND BCREATOR = 'SYSIBM'
AND BTYPE IN ('I', 'T')
ORDER BY DNAME;
```

### End of General-use Programming Interface

### Product-sensitive Programming Interface

```
SELECT DISTINCT COLLID, NAME, VERSION
FROM SYSIBM.SYSPACKDEP, SYSIBM.SYSPACKAGE
WHERE BNAME IN ('SYSCOLAUTH',
'SYSDATABASE', 'DSNDDH01', 'DSNDDX02',
'SYSDBAUTH', 'DSNADH01', 'DSNADX01',
'SYSDBRM',
'SYSINDEXES', 'DSNDXX01', 'DSNDXX02', 'DSNDXX03',
'SYSPLAN', 'DSNPPH01',
'SYSPLANAUTH', 'DSNAPH01', 'DSNAPX01',
'SYSRESAUTH', 'DSNAGH01', 'DSNAGX01',
'SYSSTOGROUP', 'DSNSSH01',
'SYSSYNONYMS', 'DSNDYX01',
'SYSTABAUTH', 'DSNATX01', 'DSNATX02', 'DSNATX03',
'SYSTABLESPACE', 'DSNDSX01',
'SYSUSERAUTH', 'DSNAUH01', 'DSNAUX02')
AND LOCATION = ' '
AND BQUALIFIER = 'SYSIBM'
AND BTYPE IN ('I', 'T')
AND COLLID = DCOLLID
AND NAME = DNAME
AND CONTOKEN = DCONTOKEN
ORDER BY COLLID, NAME, VERSION;
```

These two queries are commented out in member DSNTESQ of *prefix.SDSNSAMP*. You can execute these queries from SPUFI or from a dynamic SQL program like DSNTPE2.

After migration, you can explicitly rebind these plans and packages, or let DB2 rebind them automatically. See Section 4 of *Application Programming and SQL Guide* for suggestions on rebinding these plans and packages.

---

### Migration Step 3: Check for Consistency between Catalog Tables (Optional)

To check for consistency between catalog tables, you can run the queries that are not commented out in member DSNTESQ of the *prefix.SDSNSAMP* library.

The DSNTESQ queries check the logical correctness of the DB2 catalog. You can execute the SQL statements in DSNTESQ from SPUFI or from a dynamic SQL program like DSNTPE2.

Before you run these queries, you should have already run the DSN1CHKR utility and the CHECK INDEX utility in migration step 1.

You can execute the queries on the actual catalog tables or on “mirror” copies of the catalog. If you run the queries on the copies, use the comment lines in member DSNTESQ for guidance. Executing queries on copies reduces contention on the catalog.

---

### Migration Step 4: Image Copy Directory and Catalog in Case of Fallback: DSNTIJIC

#### Attention:

You need to create a copy of your DB2 for MVS/ESA Version 4 catalog and directory for backup purposes. DB2 starts if you do not create this copy, but if errors in the catalog or directory require you to fall back to DB2 for MVS/ESA Version 4, you risk losing some of your tables and data.

To create a copy of DB2 for MVS/ESA Version 4 catalog and directory, run DB2 for MVS/ESA Version 4 job DSNTIJIC. Before you run DSNTIJIC, examine the job for the following:

- The tape unit name. The job lists the tape unit name as TAPE. If this is incorrect for your site, correct it. The name TAPE is also used as the unit name for the default archive log data sets.
- Expiration date or retention period. You can add a retention period or an expiration date to the job.
- The USER option on the JOB statement. Make sure the user is authorized. This must be the same user you specified for either SYSTEM ADMIN 1 or SYSTEM ADMIN 2 on installation panel DSNTIPP.

Job DSNTIJIC contains a list of all the DB2 directory and catalog table spaces. When you run DSNTIJIC, it invokes the DB2 image copy utility to copy these table spaces to tape. The copied table spaces allow you to recover the DB2 catalog and directory in case of a failure.

If job DSNTIJIC fails or abends, verify that there are no problems with the tape setup for image copy. If this does not fix the problem, examine the log for problems. For instance, be sure that there are no incorrect size or I/O errors.

After migration, periodically run the DB2 for OS/390 Version 5 job DSNTIJIC against the Version 5 directory and catalog—perhaps daily or weekly. This reduces the amount of time required for recovering the DB2 directory or catalog. The copied data and log data sets are needed for recovery.

If you are remigrating, you need to do one of the following:

- Change the names of the data sets in which the new image copies will reside. (Migration image copies use the current data set names.)
- Run the MODIFY utility to remove the migration image copies. If you select this option, make sure you are familiar with the MODIFY utility. See *Utility Guide and Reference* for more information.

If DSNTIJIC has been modified to copy table spaces to DASD instead of tape, the job is limited to two DASD volumes. To change the number of DASD volumes, the job needs to be modified again using volume serial numbers instead of using `VOL=REF=*.jobstep`.

---

## Migration Step 5: Connect DB2 to TSO

Access to TSO is required to support the interactive component of DB2 (DB2I) and to allow batch applications to access DB2 when those batch programs are executed under the TSO terminal monitor program (TMP).

To attach DB2 to TSO, you must do the following:

- Make DB2 load modules available to TSO and batch users
- Make DB2 CLISTs available to TSO and batch users
- Make panels, messages, and load modules available to ISPF and TSO.

These tasks are described in the following sections.

Save your TSO LOGON procedures and JCL from DB2 for MVS/ESA Version 4 in case you need to fall back from DB2 for OS/390 Version 5.

### Making DB2 Load Modules Available to TSO and Batch Users

If you included `prefix.SDSNEXIT` and `prefix.SDSNLOAD` in your LNKLISTxx, you can skip this step.

If you have not included `prefix.SDSNEXIT` and `prefix.SDSNLOAD` in your LNKLISTxx, you must add JOBLIB or STEPLIB statements to your logon procedures and JCL to ensure that you access the Version 5 load modules. `prefix.SDSNEXIT` should be included before `prefix.SDSNLOAD` in your JOBLIB or STEPLIB concatenations.

You can attach to multiple releases of DB2 with your existing TSO or CAF logon procedures, without changing the load libraries for your applications. After you migrate completely to the latest level of DB2, you **MUST** update those procedures and jobs to point to the latest level of DB2 load libraries.

## Making DB2 CLISTs Available to TSO and Batch Users: DSNTIJVC

Tailoring changes can modify these CLISTs: DSNEMC01, DSNH, DSNU, and DSNHC. The DSNTINST CLIST reads these CLISTs from *prefix.SDSNCLST*, edits them, and places them in *prefix.NEW.SDSNTEMP*. You can modify the default values. Refer to “Completing the CLIST Processing” on page 2-219 for information on the items you can modify.

Job DSNTIJVC merges the tailored CLISTs from *prefix.NEW.SDSNTEMP* with unchanged CLISTs from *prefix.SDSNCLST*, and it places all CLISTs in *prefix.NEW.SDSNCLST*. It also converts the DB2 CLISTs from a fixed-block record format to a variable-blocked format, with a record length of 84 and a block size of 3120, if wanted.

**If you use fixed-block CLIST libraries**, modify the DSNTIJVC job as follows:

- Change the SYSIN DD to DUMMY
- Change the allocation of *prefix.SDSNCLST* to match the data control block (DCB) attributes of your other CLIST libraries.

A CLIST that has been converted from fixed block to variable block cannot be used as input to the DSNTINST CLIST; use the unedited version of the SDSNCLST data set, as created by SMP.

To make the CLISTs available to TSO and batch users, you must either concatenate *prefix.NEW.SDSNCLST* with your existing CLIST libraries or copy *prefix.NEW.SDSNCLST* into an existing CLIST library.

If you need to rerun this job, first delete data set *prefix.NEW.SDSNCLST*, which is created by this job.

When corrective service is applied to a CLIST, SMP/E changes only the *prefix.SDSNCLST* data set. You need to redo any record format changes and reapply any needed tailoring. You also need to move the CLIST to *prefix.NEW.SDSNCLST*. Corrective service (program temporary fixes) for these CLISTs is sent with ++HOLD statements, noting that this additional work can be required.

## Making Panels, Messages, and Load Modules Available to ISPF and TSO

You must concatenate the DB2 ISPF libraries with the ISPPLIB, ISPMLIB, and ISPSLIB DD statements in your logon procedures and in any of your CLISTs where they might be allocated. These libraries are *prefix.SDSNSPFP*, *prefix.SDSNSPFM*, *prefix.SDSNSPFS*, and *prefix.SDSNPFPE* if you are using English or *prefix.SDSNPFPK* if you are using Kanji DB2I panels. If you are using Online Help, include *prefix.SDSNSPFT* for ISPTLIB.

DB2I uses the ISPF PROFILE and SHARED variable pools for most panel variable fields. This makes it easier to reenter a panel when panel variables have previously been specified. For the DB2 subcommands that permit LISTS of plan names,

package names, DBRMs, and ENABLE and DISABLE statements, DB2I provides ISPF tables to contain all the user-specified variables for these subcommand keywords.

DB2I creates and maintains a set of ISPF tables in a user-defined TSO data set that is allocated to a ddname of DSNETBLS. The DB2I-generated tables in this library are DSNCONNS, DSNDBRMS, and DSNPLPKN. Table 66 shows the library table member names and their contents.

Table 66. The DB2 ISPF Table Library

|          |                                                                                                 |
|----------|-------------------------------------------------------------------------------------------------|
| DSNCONNS | ENABLE/DISABLE connection type and connection name variables referenced by plan or package name |
| DSNDBRMS | Subcommand DBRM member and LIBRARY name variables referenced by plan name                       |
| DSNPLPKN | Package list variables referenced by package name                                               |

When allocating this data set, the following DCB attributes must be assigned:

```
DSORG(P0) RECFM(F B) LRECL(80) BLKSIZE(n*LRECL)
```

where n is any integer.

The following example shows how you might set up an ALLOCATE statement to create the data set:

```
ALLOC DA(DSNSPFT) NEW SP(1 1) TR DIR(10) +
DSORG(P0) RECFM(F B) LRECL(80) BLKSIZE(800)
F(DSNETBLS) REUSE
```

The following example shows how you might allocate an existing data set to the DSNETBLS ddname:

```
ALLOC DA(DSNSPFT) F(DSNETBLS) REUSE
```

If you do not allocate the DSNPFT data set and connect it to ISPF, DB2I allocates a temporary data set for the ISPF table library members at DB2I startup. DB2I deletes this temporary data set when the ISPF session is terminated.

# DB2I uses ISPF table services to maintain individual ISPF tables within the  
# DSNETBLS data set. For performance reasons, ISPF keeps this table library in an  
# open state once an individual table has been updated. Attempts to close this data  
# set using the TSO FREE command will result in error message IKJ568611.

# For additional information on this TSO error message and how to close this data  
# set, refer to *ISPF Version 4 Messages and Codes*.

If you want to run the ISPF/CAF sample application provided with DB2, be sure that the data set *prefix.RUNLIB.LOAD* is included in the STEPLIB concatenation of the logon procedure or in the ISPLLIB concatenation list. For more information about the ISPF/CAF sample application, see "Running Dynamic SQL and the ISPF/CAF Application" on page 2-321.

Refer to 2-276 for more information on using your TSO and CAF logon procedures.

---

## Migration Step 6: Connect IMS to DB2 (Optional)

Connecting DB2 to IMS requires coordination with your IMS support group. To connect the IMS attachment facility, you must:

- Make DB2 load modules available to IMS
- Define DB2 to IMS
- Define new programs and transactions to IMS
- Prepare IMS applications for DB2.

Depending on your site, you might also need to:

- Define DB2 plans
- Generate a user language interface.

These tasks are described in “Chapter 2-10. Connecting the IMS Attachment Facility” on page 2-371. This chapter also covers the requirements from a DB2 perspective and refers to IMS books for specific IMS information.

---

## Migration Step 7: Connect CICS to DB2 (Optional)

Connecting DB2 to CICS requires that you regenerate several CICS tables with additional entries. A macro is supplied with DB2 to define the connection between CICS and DB2 using a resource control table (RCT).

You must reassemble the RCT using the DB2 for OS/390 Version 5 libraries, unless you have CICS Version 4, in which case you will use the RCT in the CICS Version 4 macro library (CICS410.SDFHMAC).

Be sure that you coordinate the attachment facility connection with your CICS support group. To connect the CICS attachment facility, you must do the following:

#  
#

- Add CICS routines to DB2 load modules by running job DSNTIJSU (Job DSNTIJSU is not required for CICS Version 4 and subsequent releases.)
- Recalculate space requirements for the CICS attachment facility
- Define your CICS attachment facility parameters using the RCT
- Update the CICS system tables
- Update the CICS initialization JCL
- Coordinate DB2 and CICS security if necessary
- Prepare new CICS applications for DB2 if necessary.

These tasks are discussed in “Chapter 2-11. Connecting the CICS Attachment Facility” on page 2-379.

If you use CICS 3.3, you must manually add the TASKDATALOC(BELOW) parameter to the transaction definition. For more information, see “Transaction Entries” on page 2-387.

---

## Migration Step 8: Stop DB2 for MVS/ESA Version 4 Activity

Before making DB2 for OS/390 Version 5 operational, ensure that all work is stopped on DB2 for MVS/ESA Version 4, including data sharing members if you have enabled data sharing. If you do not stop work on Version 4, fallback procedures may fail.

To stop work on Version 4, perform the following steps:

1. Issue the command:

```
-DSN1 STOP DB2 MODE(QUIESCE)
```

where *-DSN1* is the subsystem command prefix defined for DB2.

The QUIESCE keyword allows DB2 to complete processing of currently executing programs. This can require some processing time.

2. Issue the command:

```
-DSN1 START DB2 ACCESS(MAINT)
```

This allows only the install-defined system administrators and system operators to access DB2.

If DB2 does not start properly, it usually abends with a reason code indicating where the error occurred. To find the error, check the set of definitions for the associated resource. For example, if the bootstrap data set (BSDS) does not match the subsystem parameter values, make sure the correct jobs were run for DSNTIJUZ. Make sure you started DB2 with the correct subsystem parameter option.

3. Make sure all work is complete.

- Make sure no units of recovery remain. Issue the command:

```
-DSN1 DISPLAY THREAD(*) TYPE(*)
```

Then use `-DSN1RECOVER INDOUBT` for any indoubt threads.

- Make sure no utility work remains. Issue the command:

```
-DSN1 DISPLAY UTILITY(*)
```

Then, either allow utilities to complete before proceeding, or issue the command:

```
-DSN1 TERM UTILITY(*)
```

to stop all utility processing.

- Make sure no table spaces and index spaces in the DB2 directory (DSNDB01) or the DB2 catalog (DSNDB06) have write error ranges or deferred restart states. You can determine existing restrictions by issuing this command:

```
-DSN1 DISPLAY DATABASE(*) SPACENAM(*) RESTRICT
```

A user with system administrator or system operator authority specified on installation panel DSNTIPP, must enter this command.

Recover any table spaces and index spaces with write error ranges or deferred restart states.

4. To stop DB2, issue the command:

```
-DSN1 STOP DB2 MODE(QUIESCE)
```

A user with install-defined system administrator or system operator authority must enter this command.

---

## Migration Step 9: Back up Your DB2 for MVS/ESA Version 4 Volumes (Optional)

At this point, you can back up your DB2 for MVS/ESA Version 4 subsystem. To do this, take dumps of the DB2 subsystem data sets. You also can take dumps of the SMP/E data sets and the DB2 distribution and target libraries.

---

## Migration Step 10: Define DB2 Initialization Parameter: DSNTIJUZ

DSNTIJUZ generates the DB2 data-only subsystem parameter module, DSNZPxxx. The subsystem parameter module consists of the expansion of seven macros that contain the DB2 execution-time parameters that you selected using the ISPF panels. The names of these macros are DSN6ARVP, DSN6ENV, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSP.

Save your DB2 for MVS/ESA Version 4 subsystem parameter to have it available in case you need to fall back.

The DSNTINST CLIST performs calculations using the values you specified for some of the parameter values you entered on the panels. These calculations appear in the macro descriptions.

### DSNTIJUZ Actions

Besides defining the subsystem parameter module, job DSNTIJUZ does the following:

- Link-edits the assembled subsystem parameter module into the *prefix.SDSNEXIT* library.
- Uses the assembler, the DSNHDECM macro, and SMP/E to update values you specified on installation panels DSNTIPF and DSNTIP4 and your subsystem name. The information is placed into a data-only load module, DSNHDECP, which resides in *prefix.SDSNEXIT*.
- Uses SMP/E in step DSNTIMQ to read in the edited version of DSNTIJUZ. This is required to pick up the appropriate includes and library names. After the initial run of step DSNTIMQ, it is only required when changes have been made to DSNHDECP.
- Uses JCLIN to ensure that macro maintenance is placed into all the needed load modules.

### Additional Steps

1. If you added a STEPLIB statement to the DB2 start procedures ahead of *prefix.SDSNEXIT* and *prefix.SDSNLOAD*, you can move the SYSLMOD output to that library.
2. If you changed the prefix for the DB2 distribution libraries, edit DSNTIJUZ to correct the data set names.
3. If you have not run the SMP/E ACCEPT job (DSNTIJAC) of FMID HDB5510, you must edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZQ. This ensures that member DSNARIB is linked with DSNHDECP.



The linkage editor issues a return code of 8 along with message IEW0342 for the following CSECTs:

|          |          |          |          |
|----------|----------|----------|----------|
| DSNFSYSP | DSNJARVP | DSNJLOGP | DSNTSPRM |
| DSNVDIR1 | DSNZMSTR | DSN3DIR1 |          |

## Considerations for the BSDS

If your DB2 for MVS/ESA Version 4 system has only one BSDS, you must either:

- Manually add TWOBSDS=NO in the DSN6LOGP macro in job DSNTIJUZ
- Add another BSDS to DB2 before you migrate.

We recommend adding a second BSDS because having two BSDSs makes recovery much easier in most situations. In cases that normally require recovery and restart, a second BSDS allows you to continue working. Also, the storage required is small and the data set is relatively inactive.

To add a second BSDS:

1. Change your subsystem parameter to TWOBSDS=YES using job DSNTIJUZ.
2. Define a second BSDS using as an example the VSAM BSDS definition in job DSNTIJIN.
3. Add a //BSDS2 DD statement to the DSNMSTR DB2 startup procedure.
4. Execute the -RECOVER BSDS command to establish dual BSDS. For information on the -RECOVER BSDS command, see Section 4 (Volume 1) of *Administration Guide*.

You might receive message GIM65001W when running steps DSNTLOG and DSNTIMQ or receive a return code of 4 when running step DSNTIMQ. You can ignore these messages.

If DSNTIJUZ fails or abends, correct the problem and rerun the job, using the same subsystem parameter name.

---

## Migration Step 11: Establish Subsystem Security (Optional)

DB2 includes means for controlling access to data within DB2. It also works together with outside security systems, such as RACF, that control access to the DB2 system. See Section 3 (Volume 1) of *Administration Guide* for suggestions and instructions for including DB2 in your security system.

Because your Version 5 system reuses the data objects from your Version 4 system, you have probably already supplied the protection those objects need. However, you probably want to protect the new (Version 5) DB2 for OS/390 data objects.

---

## Migration Step 12: Define DB2 for OS/390 Version 5 to MVS: DSNTIJMV

This job does some of the steps required to identify DB2 to MVS, including updating members of SYS1.PARMLIB and SYS1.PROCLIB. This job renames your DB2 for MVS/ESA Version 4 procedures so they do not conflict with Version 5 procedures.

Because different sites have different requirements for identifying DB2 to MVS, it is not possible for DSNTIJMV to anticipate all the updates necessary. For this reason, the updates that job DSNTIJMV makes in SYS1.PARMLIB and SYS1.PROCLIB are incomplete. You might have additional procedures of your own to rename, or you could provide procedures for both releases, using alias names to indicate the current release. You can complete these updates either by making the updates directly in SYS1.PARMLIB and SYS1.PROCLIB or by editing DSNTIJMV.

For SYS1.PROCLIB, we recommend that you submit the procedure-update section of DSNTIJMV, as necessary. However, before you make the updates, read this section of the chapter and examine DSNTIJMV to study the updates it makes. We recommend that you edit the updates directly in SYS1.PARMLIB instead of submitting the updates in the DSNTIJMV step.

Whether you make the updates directly or edit DSNTIJMV to make the updates, you might first want to review “Choosing Link List Options” on page 2-74.

## DSNTIJMV Actions

1. Job DSNTIJMV updates the following SYS1.PARMLIB members:

- IEFSSNxx

This member contains an entry for every MVS subsystem. Unless you change the DB2 subsystem name or the DB2 command prefix, you do not need to change this member. If you change the subsystem name or the command prefix, either change the current member or create a new member.

You must place the line describing SMS first (if you are using SMS), followed by the JES subsystem, then the DB2 line anywhere after the JES line.

- IEAAPFxx or PROGxx

Job DSNTIJMV updates IEAAPFxx to include the DB2 program libraries (*prefix.SDSNEXIT*, *prefix.SDSNLOAD*, *prefix.SDXRRESL*, and *prefix.SDSNLINK*) as libraries authorized using the authorized program facility (APF).

All libraries concatenated with *prefix.SDSNEXIT* and *prefix.SDSNLOAD* in STEPLIB and JOBLIB statements must be APF-authorized.

If you are using MVS/ESA Version 4 Release 3, you might be using the PROGxx member instead of the IEAAPFxx member. If so, you have to update this member manually—job DSNTIJMV does not edit it.

- LNKLSTxx

Job DSNTIJMV updates this member to include the DB2 load module library, *prefix.SDSNLINK*, in the LNKLSTxx. If you moved the modules from *prefix.SDSNLINK* into another library, edit DSNTIJMV to include that library in LNKLSTxx. If you have combined *prefix.SDSNLINK* and *prefix.SDSNLOAD* into one library, edit DSNTIJMV to include the combined library in LNKLSTxx. See *OS/390 MVS Initialization and Tuning Guide* for restrictions on data sets that are concatenated in LNKLST.

**You can do additional editing** for the SYS1.PARMLIB updates. If you are editing DSNTIJMV rather than making the changes directly, you have a choice: You can either include your additional entries for the SYS1.PARMLIB members

(IEAAPFxx and LNKLSTxx) at the end of the existing list of entries, or you can place them earlier in the list.

If you include these entries at the end of the existing SYS1.PARMLIB list, make sure there are commas (the continuation character) delimiting each entry except the last.

**ECSA size** is another SYS1.PARMLIB change to consider at this time. It is specified in the CSA parameter of the IEASYS00 parameter. Be sure that you have specified an adequate size for this subsystem (generally 2MB plus the MAXIMUM ECSA value on installation panel DSNTIPJ if the CROSS MEMORY value is NO).

The **IOP parameter** is another SYS1.PARMLIB change to consider at this time. If you are running with MVS/SP Version 4 Release 3 and DFSMS/MVS Version 1 Release 1, DB2 can schedule I/O priority. To enable this, you must:

- Use the IOP parameter to set the I/O priority for the address space of a performance group. The IOP parameter is in the IEAIPSxx member of SYS1.PARMLIB.
  - Enable MVS I/O priority scheduling by specifying IOQ=PRTY in the IEAIPSxx member of SYS1.PARMLIB.
2. Job DSNTIJMV renames your DB2 for MVS/ESA Version 4 procedures so they are not replaced by DB2 for OS/390 Version 5 procedures.
  3. DSNTIJMV updates SYS1.PROCLIB to include the following Version 5 procedures:
    - System services address space startup procedure (xxxxMSTR)
    - Database services address space startup procedure (xxxxDBM1)
    - Distributed data facility address space startup procedure (xxxxDIST)
    - Stored procedures address space startup procedure (xxxxSPAS)
    - WLM sample procedure for stored procedures
    - IRLM address space startup procedure (IRLMPROC)
    - Program preparation procedures
    - Utilities procedure (DSNUPROC).

If you specified a suffix on panel DSNTIPA1, that suffix is appended to data sets &USER..DBRMLIB.DATA.suffix, &USER..RUNLIB.LOAD.suffix, and &USER..SRCLIB.DATA.suffix. To override these data set names, you must edit the updates to SYS1.PROCLIB.

## Completing the Step

During migration, DB2 for OS/390 Version 5 procedures replace your DB2 for MVS/ESA Version 4 procedures (which are renamed). If you changed the DB2 subsystem name, the name of the DB2 address space startup procedures also change. If you made any changes to your Version 4 procedures (such as data set names), make similar changes to the Version 5 procedures.

Before bringing up DB2, check the private area sizes in the SYS1.PROCLIB update section to be sure that you have enough user private area.

Also, examine the size of the private area on the DB2 start procedures. If necessary, modify them to satisfy the requirements for EDM pool size, buffers, numbers of data sets open, and the amount of available private address space.

For more information about private address spaces, see “Working Storage Calculation” on page 2-46.

If job DSNTIJMV runs successfully, it produces return codes of 0. Because a rename can fail without setting the return code, check all renames.

---

## Migration Step 13: Define System Data Sets: DSNTIJIN

Job DSNTIJIN defines these non-VSAM data sets:

```
prefix.SRCLIB.DATA
prefix.RUNLIB.LOAD
prefix.DBRMLIB.DATA
```

The job also defines the VSAM data sets for the table spaces and indexes of the catalog and directory. For recovery purposes, it is best to keep system data sets on different DASD volumes. Because these data sets are in use frequently, do not migrate them with DFSMSHsm.

If DSNTIJIN runs successfully, it produces return codes of 0 for all steps. Check any VSAM messages carefully.

If job DSNTIJIN fails or abends, delete the allocated non-VSAM data sets, examine the VSAM messages, and correct any indicated problems with DFSMSdfp, then rerun job DSNTIJIN.

---

## Migration Step 14: Define User Authorization Exit Routines: DSNTIJEX

Job DSNTIJEX builds the sample authorization exits, DSN3@SGN and DSN3@ATH, from the source code in *prefix.SDSNSAMP* and places them in the *prefix.SDSNEXIT* library. You can modify DSNX@XAC, the access control authorization exit, and use DSNTIJEX to assemble and linkedit it. This exit allows you to bypass some or most of DB2 authorization checking and to specify your own authorization checking. For more information on this access control authorization exit, see Appendix B of *Administration Guide*. The DB2 CLIST tailors the JCL in DSNTIJEX to meet your site's environment.

The sample authorization exits are not the same as the default authorization exits supplied by DB2. By implementing the sample authorization exits, you can provide group names as secondary authorization IDs. For information on writing exit routines, see Appendix B of *Administration Guide*. For more information on controlling data access, see Section 3 (Volume 1) of *Administration Guide*.

You have the following options regarding exit routines:

- To use the default authorizations, skip job DSNTIJEX.
- To use the sample authorization exits, run job DSNTIJEX.
- To use your own authorization exit routines, modify job DSNTIJEX to reference the correct library, then run it.

If job DSNTIJEX runs successfully, it produces a return code of 0.

If job DSNTIJEX fails or abends, correct the problem and rerun the job.

---

## Migration Step 15: IPL MVS

The load module library SDSNLINK contains the early code. If all of the required maintenance has been applied to your system, the early code is *upward* compatible with DB2 for OS/390 Version 5. Be sure that the early code pre-conditioning PTFs have been installed on your system before you migrate. The Version 5 early code is *downward* compatible with DB2 for MVS/ESA Version 4.

Provided that you are at the appropriate service level for Version 4, you can plan ahead, do PARMLIB updates (necessary at least to update the APF authorization list), and IPL MVS whenever convenient, before you begin your migration. The MVS IPL is necessary because migration job DSNTIJMV makes changes to SYS1.PARMLIB that are not recognized by MVS until the next IPL. The DSNTIJMV job makes the following changes to the SYS1.PARMLIB library:

- Creates new subsystem definitions in the IEFSSNxx member
- Creates new APF libraries in the IEAAPFxx member
- Creates new load module libraries in the LNKLSTxx member.

Complete these changes before performing the IPL.

You must IPL before or during migration, but IPLs are not necessary for fallback or remigration. For more information, refer to “Migration Step 12: Define DB2 for OS/390 Version 5 to MVS: DSNTIJMV” on page 2-281 and “Choosing Link List Options” on page 2-74.

After you IPL MVS, message DSN3100I appears on the MVS console, stating that DB2 is ready for the START command.

---

## Migration Step 16: Start DB2 for OS/390 Version 5

Perform the following steps to start DB2 for OS/390 Version 5:

1. Start the IRLM.

If you have not requested that DB2 automatically start the IRLM, you should start it before you start DB2. Use the command:

```
START irlmproc
```

where *irlmproc* is the name you assigned to the IRLM startup procedure. This is the value you specified for the PROC NAME option on installation panel DSNTIPI.

If you specified YES for the AUTO START option on installation panel DSNTIPI, DB2 starts the IRLM automatically.

2. Start DB2 from the MVS console with the command:

```
-DSN1 START DB2 PARM(DSNZPxxx)
```

where -DSN1 is the subsystem command prefix you defined for DB2, and *DSNZPxxx* is the name of the DB2 initialization parameter module. If you omit the PARM parameter, the name used is the one you specified in the field PARAMETER MODULE on panel DSNTIPI. If you did not specify a parameter module on panel DSNTIPI, it defaults to DSNZPARM.

If DB2 starts successfully, 2 to 5 address spaces also start. These address spaces are *ssnmMSTR* and *ssnmDBM1*, and possibly *ssnmDIST*, *ssnmSPAS*,

and *irlmproc*, where *ssnm* is the DB2 subsystem name and *irlmproc* is the IRLM procedure name.

If DB2 starts successfully, the series of RESTART messages you receive concludes with these two messages:

```
DSNR002I RESTART COMPLETED
DSN9022I DSNYASCP '-DSN1 START DB2' NORMAL COMPLETION
```

In the next step you migrate the DB2 catalog. Before the catalog is migrated, some catalog or directory table spaces are restricted. The following messages may occur during startup and can be ignored because the catalog and directory table spaces are restricted:

- DSNT501I with reason code 00C900A6
- DSNL700I with reason code 00C900A6 (if DDF is auto started)
- abend 04E with reason code 00E70014 (during DDL registration)

You can determine existing restrictions by issuing this command after you start DB2:

```
-DSN1 DISPLAY DATABASE(*) SPACENAM(*) RESTRICT
```

The above command might also produce message DSNT501I with reason code 00C900A6 .

If DB2 does not start properly, it usually abends with a reason code indicating where the error occurred. To find the error, check the set of definitions for the associated resource. A common cause of startup failure is that the BSDS does not match the subsystem parameter values; be sure that the correct job was run for DSNTIJUZ. Also, check that the subsystem parameter member you specified (or allowed to default) when you started DB2 is the one built by the DSNTIJUZ job. Check the JCL for the DB2 start procedure.

3. Optionally, start TSO. If you want to use the TSO SUBMIT command to do housekeeping and migration verification, you must start TSO (if it is not already started).

---

## Migration Step 17: Tailor DB2 for OS/390 Version 5 Catalog: DSNTIJTC

DSNTIJTC invokes the CATMAINT utility to migrate your DB2 for MVS/ESA Version 4 catalog to the DB2 for OS/390 Version 5 catalog. Before running this job, ensure you have enough DASD space as described in "DASD Requirements for the Work File Database" on page 2-21 or the job will fail. See Table 65 on page 2-268 for a list of new and changed indexes that might affect your work file database space needs. DSNTIJTC creates new catalog and directory objects, adds columns to existing catalog tables and creates and updates indexes on the catalog tables to accommodate new Version 5 objects. All the indexes are created or updated sequentially during the execution of DSNTIJTC.

A new status message, DSNU777I, is issued at several points during the migration process to indicate migration progress. New diagnostic error messages are issued when CATMAINT processing fails. If a problem is found during the SQL processing phase of migration then message DSNU778I is issued. If non-supported functions are encountered such as type 1 indexes then message DSNU776I is issued. See the message descriptions in *Messages and Codes* for details.

#  
#

If job DSNTIJTC fails, save the output and verify that you are at the correct maintenance level. If you are not, you need to install the appropriate maintenance. If you are, correct the problem and run the entire job again. If you run the job again and the job still fails, return to Version 4 by restoring your Version 4 catalog, directory and BSDS. When you migrate, run the job again.

To execute DSNTIJTC, you must have installation SYSADM authority.

---

## Migration Step 18: Ensure There Are No Problems with the Catalog (Optional)

We recommend that you check the integrity of your DB2 for OS/390 Version 5 catalog by following, in any order, these steps.

- Run CHECK INDEX on all the indexes in the catalog and directory.
- Run link checker (DSN1CHKR) to ensure that there are no existing broken links. See “Migration Step 1: Run Link Checker on DB2 for MVS/ESA Version 4 Table Spaces (optional)” on page 2-272 for more information.
- Run the queries in member DSNTESQ of *prefix.SDSNSAMP*. Because SPUFI is not bound yet, you cannot use SPUFI to run these queries. One alternative is to use the Version 4 DSNTPE2 program to execute the queries.
- Run the DSN1COPY utility with the CHECK option on the catalog table spaces.

---

## Migration Step 19: Prepare Dynamic SQL Program: DSNTIJTM

DSNTIJTM assembles, link-edits, binds, and runs DSNTIAD, a program that processes certain SQL statements dynamically. For data sharing, the job is only edited for the first member to migrate.

---

## Migration Step 20: Bind DB2 objects and User Maintained Database Activity: DSNTIJSG

#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#

Your flagger, schema processor, SPUFI and DCLGEN plans from DB2 for MVS/ESA Version 4 are usable in Version 5 only if they were bound as packages or if PTF UN98205 was applied to Version 4. If they were bound as plans, you must have already converted them to packages **before** migrating to Version 5. See UN98205 for more information. In migration mode, the DSNTIJSG job rebinds only the packages for flagger, schema processor, SPUFI and DCLGEN.

In migraton mode, job DSNTIJSG does the following:

- rebinds IBM-defined packages
- alters the RLST
- defines the DB2-supplied stored procedures:
  - Utility invocation stored procedure
  - Subsystem parameter stored procedure

See Appendix B of *Utility Guide and Reference* for information on stored procedures that perform utility functions. Job DSNTIJSG also copies the rows of the previous communication database (CDB), which was a user defined database prior to DB2 Version 5, into the new CDB. This puts the CDB tables in the proper

# format for the DB2 catalog. The table names in the CDB have changed and new  
# tables have been added in support of TCP/IP. We recommend that you update both  
# your current CDB and the old CDB if the CDB information for VTAM partners is  
# updated. The old CDB tables are used if you fall back to a previous release of DB2.  
# See “Step 4: Populate the Communications Database” on page 3-20 and “Step 4:  
# Populate the Communications Database” on page 3-56 for more information.

# If DSNTIJSJG runs successfully, it produces a return code of 0. It can also produce  
# a return code of 4 since a step within this job attempts to delete a row from a table  
# that might not exist at the time this job is run. There is a bind warning for each  
# plan. Expect the following messages:

```
DSNE932I WARNING, ONLY IBM-SUPPLIED PLAN NAMES SHOULD BEGIN WITH DSN
DSNE932I WARNING, ONLY IBM-SUPPLIED PACKAGE-IDS SHOULD BEGIN WITH DSN
DSNE932I WARNING, ONLY IBM-SUPPLIED COLLECTION-IDS SHOULD BEGIN WITH DSN
```

# When you migrate the first member of the data sharing group to Version 5,  
# DSNTIJSJG rebinds SPUFI in Version 5. The Version 4 members cannot use a  
# Version 5 SPUFI. If you attempt to run an SQL statement in a data sharing member  
# at Version 4 with SPUFI at Version 5, expect the following messages:

```
DSNT408I SQLCODE = -904, ERROR: UNSUCCESSFUL EXECUTION
 CAUSED BY AN UNAVAILABLE RESOURCE. REASON 00E7009E
DSNT418I SQLSTATE = 57011 SQLSTATE RETURN CODE
```

If job DSNTIJSJG fails or abends, be sure that the user specified on the JOB statement is authorized. Use the name you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP. (The RESTART parameter on the JOB statement can be useful.)

Correct any other problems, uncomment the DSNTIJSJG restart step at the beginning of the job, and rerun DSNTIJSJG. If you encounter resource shortages, review the parameters in job DSNTIJUZ, making any necessary modifications. Then stop DB2, rerun DSNTIJUZ, start DB2, and rerun DSNTIJSJG from the last successful step.

---

## # Migration Step 21: Bind the Packages for DB2 REXX Language Support: DSNTIJRX

# Before you can use DB2 REXX Language Support, you must bind DB2 packages  
# that DB2 REXX Language Support uses. Run job DSNTIJRX to do this. Before  
# you run DSNTIJRX, make the following changes:

- # • Add a job statement.
- # • Change DSN SYSTEM(DSN) to DSN SYSTEM(ssid), where ssid is the name of the  
# DB2 subsystem on which you will use DB2 REXX Language Support.
- # • Change all instances of DSN!!0 to your DB2 data set name prefix.
- # • Change all instances of DSNTIA!! to the plan name for the DSNTIAD program.



---

## Migration Step 22: Image Copy DB2 for OS/390 Version 5 Catalog: DSNTIJIC

Create a copy of the Version 5 DB2 for OS/390 catalog and directory for backup purposes. See “Migration Step 4: Image Copy Directory and Catalog in Case of Fallback: DSNTIJIC” on page 2-274 for information on job DSNTIJIC.

---

## Migration Step 23: Verify Your DB2 for OS/390 Version 5 System

Run your DB2 for MVS/ESA Version 4 sample jobs on your DB2 for OS/390 Version 5 subsystem, and then run the Version 5 sample jobs on your Version 5 subsystem.

### Running the DB2 for MVS/ESA Version 4 Sample Jobs (Optional)

If all of the local DB2 objects from Version 4 still exist (that is, if you have not run job DSNTTEJ0), follow the procedure below to test migration to Version 5:

1. Do not run job DSNTTEJ0.
2. Change the JOBLIB statements to point to *prefix*.SDSNLOAD.
3. Be sure the DSN8EAE1 module you created when you originally ran the Version 4 sample jobs is copied to *prefix*.SDSNEXIT. DSN8EAE1 is an EDITPROC used by the employee sample table.
4. Execute the RUN statements in step 2 of job DSNTTEJ1P. Performing this step runs job DSNTTEP2 from Version 4.
5. Test migration of the IVP phase 2 applications from Version 4:
  - a. DSNTTEJ2A: Perform all but the first two steps of job DSNTTEJ2A.  
Expect a return code of 4 because table spaces DSN8D41U.NEWDEPT and DSN8D41U.NEWPHONE are placed in copy pending states.
  - b. DSNTTEJ2C: Execute only the RUN PROGRAM(DSN8BC3) PLAN (DSN8BH41) statement in step PH02CS04.
  - c. DSNTTEJ2D: Execute only the RUN PROGRAM(DSN8BD3) PLAN (DSN8BD41) statement in step PH02DS04.
  - d. DSNTTEJ2F: Execute only the RUN PROGRAM(DSN8BF3) PLAN (DSN8BF41) statement in step PH02FS04.
  - e. DSNTTEJ2P: Run only step PH02PS05.
6. Test migration of the IVP phase 3 applications from Version 4:
  - a. Do not run job DSNTTEJ3C
  - b. Edit the Version 4 SDSNSAMP library member DSNTTESA and insert the following at the beginning:  
DROP TABLESPACE TEMPSPAC;
  - c. Run step 4 of job DSNTTEJ3P. Expect a return code of 8 because of the rollback processing in this step.
  - d. If you want to test the DB2 Version 4's ISPF/CAF applications under Version 5, you must place the Version 4 SDSNSPFP panel library ahead of the Version 5 SDSNSPFP panel library in the ISPLLIB concatenation. This is necessary in order for the plans migrated from Version 4 to be used. Be

```
sure to remove the Version 4 SDSNSPFP library from your ISPLLIB
concatenation when you are finished testing the Version 4 IVP applications
under Version 5.
#
See the DB2 Version 4 Installation Guide for directions on how to start and
run the Version 4 ISPF/CAF applications.
```

7. Do not run any other Version 4 IVP jobs.

```
If you have already deleted the DB2 sample objects, run phases 1 through 3 of the
DB2 for MVS/ESA Version 4 DSNTEJxx jobs. Specifically, you need to run jobs
DSNTEJ1, DSNTEJ1P, DSNTEJ2A, DSNTEJ2C, DSNTEJ2D, DSNTEJ2E,
DSNTEJ2F, DSNTEJ2P, DSNTEJ3C, and DSNTEJ3P. Your JOBLIB statements
should point to the Version 5 libraries.
```

## Running the Version 5 Sample Jobs

The detailed instructions for running your Version 5 sample jobs on your Version 5 subsystem are described in “Chapter 2-9. Verifying with the Sample Applications” on page 2-305.

---

## Migration Step 24: Make Adjustments for Release Incompatibilities

These changes might affect your DB2 operations after migrating to Version 5.

```
Restrictions in STOGROUP VOLUMES clause enforced: The PTF for PQ01849
enforces a documented restriction that only a single asterisk can be specified in a
quoted string in a CREATE STOGROUP or ALTER STOGROUP volume list. Any
STOGROUP definitions with multiple asterisks in a quoted string in the VOLUMES
clause must be removed using the ALTER STOGROUP REMOVE VOLUMES
statement before applying the PTF. These illegal definitions cause certain DB2
functions to fail. You will be unable to remove these illegal definitions after applying
this PTF.
```

## Adjust Application Programs

You might need to adjust your application programs because of the following release incompatibilities.

```
SQLCODE -101: After migrating to DB2 Version 5, it is possible to receive
SQLCODE -101 on long or complicated SQL statements that previously executed
successfully. This is possible because SQL statements and DB2 internal structures
are buffered in the same local storage, and release changes in the internal
structures can result in less storage available for the SQL statements.
```

Re-write the offending SQL statements by using correlated references, or by breaking up UNIONS.

```
| New SQL reserved word, CASE: If you have columns in your applications that are
| named CASE, you should now refer to them as “CASE,” or rename your columns.
| The word CASE is now a reserved word in IBM SQL. It is a good idea to avoid any
| words that are listed as reserved.
```

```
| SQLSTATE changes: Queries that returned SQLCODE '-302' with SQLSTATE
| '22001', String too long and number too large for target, now return SQLSTATE
| '22003', Number too large for target, for numeric operations.
```

SQLCODE '-130' with SQLSTATE '22019' previously returned two conditions: *Invalid escape pattern* and *Invalid occurrence of escape character in pattern of LIKE predicate*. The second condition is now reported with SQLSTATE '22025' instead.

# **Character column length changes:** The maximum length for a character column (CHAR or VARCHAR) has changed from 254 to 255 bytes. Therefore CHAR(255) and VARCHAR(255) are considered short strings. CREATE TABLE, CREATE INDEX, and CREATE VIEW can use objects up to 255 bytes long. The limits on TYPE 2 indexes has increased to 255 however TYPE 1 indexes remain with the 254 byte limit. This means ALTER INDEX TYPE 2 to TYPE 1 will fail if the TYPE 2 index is 255 bytes long. The maximum length of literals has increased from 254 to 255 bytes. The DB2 catalog is not changed to use the longer character column length.

# **ALTER TABLE changes:** A new clause called ALTER COLUMN is added to the ALTER TABLE statement. The new clause allows the definition of an existing varchar column to be changed up to the maximum length allowed for the varchar data type. If you do not have APAR PQ20644 installed, all indexes on the table using the increased column size will be placed in recover pending state. Run the RECOVER INDEX utility to rebuild the affected indexes. **Recommendation:** Run RUNSTATS to update the DB2 catalog statistics after the indexes have been rebuilt.

| **Catalog statistic column changes:** Applications that update the statistic columns CARD, FULLKEY FIRSTKEYCARD, FREQUENCY, or COLCARD need to change to update CARDF, FULLKEYCARD, FIRSTKEYCARD, FREQUENCYF, or COLCARD, respectively.

There are new columns added and new definitions for existing columns in the catalog tables SYSCOLDIST and SYSCOLDISTSTATS. The new columns are TYPE, CARDF, NUMCOLUMNS, and COLGROUPCOLNO in both SYSCOLDIST and SYSCOLDISTSTATS. These catalog changes relate to collecting statistics on concatenated key columns of an index. See the *SQL Reference* for more details on the catalog changes.

# **Secondary quantity values from the DB2 catalog:** For DB2-managed data sets, DB2 is getting secondary quantity values from the DB2 catalog when the data set is extended. These values were previously obtained from the integrated catalog. If you change the secondary quantity in the DB2 catalog, that new value is not reflected in the integrated catalog until you run the utility REORG, LOAD REPLACE, or RECOVER.

**Trace Record Changes:** The formats of trace records for IFCIDs 22 and 221 have changed for Version 5. Those formats are not compatible with the Version 4 formats. You must change any applications that process those trace records.

**Detailed error codes for clients:** Clients that previously received reason code -904 now receive SQLCODE '-30082' if the following conditions exist:

- EXTENDED SECURITY field is YES on installation panel DSNTIPR
- VTAM Version 4 Release 4 is used on the DB2 server
- Client can accept security failure reason codes.

**Default PRELINK parameter change:** The new default for the PRELINK parameter is YES. If you use a language that supports prelinking, such as C/370 Version 2 Release 1, you must specify PRELINK(NO) if you want to avoid prelinking. For languages that do not support prelinking such as Assembler, Fortran, VS COBOL, and PL/I specify PRELINK(NO).

**Changes to DISPLAY DATABASE:** The increase in the maximum number of partitions in a partitioned tablespace from 64 to 254, changed the number of bytes used to display the part number. You should be aware that the PART field of DISPLAY DATABASE now uses three bytes instead of two bytes. The lock information is still contained within this field, but the left and right parentheses are removed and the left parenthesis is replaced with a dash.

In addition, the output that appears in the LOCKINFO column of DISPLAY DATABASE is changed to more efficiently use the 10 bytes allotted to it. The new output has this appearance:

```
LOCKINFO

Q-III,UU,D
```

where **Q** is the lock qualifier, **III** is the lock identifier, **UU** is the lock unit and **D** is the lock descriptor.

**Dynamic DELETE WHERE CURRENT OF used with CURRENT DATA(NO):** An application can use dynamic DELETE WHERE CURRENT OF on ambiguous cursors in statements that are bound with CURRENTDATA(NO). If DELETE WHERE CURRENT OF is executed on a row where an ambiguous cursor is positioned, and the statement was bound CURRENTDATA(NO), and if DB2 can determine that data has not changed since it was fetched, the delete is processed successfully. If DB2 cannot determine if the data has changed, then your application might receive SQLCODE '-510'. You should modify your applications to handle this code.

**Changes to TABLESPACES defined with LOCKSIZE ANY:** Tablespaces created before Version 3.1 with LOCKSIZE ANY and used on later releases would no longer have lock escalation until an ALTER TABLESPACE LOCKMAX n is issued on the tablespace. See APAR PQ24032 for details.

## # Changes to DISPLAY DATABASE

The increase in the maximum number of partitions in a partitioned tablespace from 64 to 254, changed the number of bytes used to display the part number. You should be aware that the PART field of DISPLAY DATABASE now uses three bytes instead of two bytes. The lock information is still contained within this field, but the left and right parentheses are removed and the left parenthesis is replaced with a dash.

In addition, the output that appears in the LOCKINFO column of DISPLAY DATABASE is changed to more efficiently use the 10 bytes allotted to it. The new output has this appearance:

```
LOCKINFO

Q-III,UU,D
```

where **Q** is the lock qualifier, **III** is the lock identifier, **UU** is the lock unit and **D** is the lock descriptor.

# In Version 4, the command DISPLAY DATABASE with the RESTRICT option listed  
# all objects in the database, whether they were in a restricted state or not. In  
# Version 5, only objects that are in a restricted state are displayed.

## SQL92 Entry Level Conformance

Several changes in SQL may affect existing queries. See *SQL Reference* for details on these changes.

## View WITH CHECK OPTION

Changes to the semantics of the WITH CHECK OPTION clause present potential problems for views created in Version 5. The following SELECT statement helps to identify such views. This statement is added to DSNTESQ as a new sample.

General-use Programming Interface

```
SELECT B.CREATOR, B.NAME, D.CREATOR, D.NAME
FROM SYSIBM.SYSVIEWS B,
 SYSIBM.SYSVIEWS D,
 SYSIBM.SYSTABAUTH A
WHERE B.CREATOR = A.SCREATOR
 AND B.NAME = A.STNAME
 AND A.GRANTEETYPE = ' '
 AND B.CHECK = 'Y'
 AND D.CHECK = 'N'
 AND D.CREATOR = A.TCREATOR
 AND D.NAME = A.TTNAME
 AND D.CREATOR = A.GRANTOR
 AND A.GRANTEE = A.GRANTOR
 AND A.UPDATEAUTH <> ' ';
```

End of General-use Programming Interface

For detailed information about changes to creating views, see Application Programming and SQL Guide.

## SELECT Privilege for UPDATE and DELETE

SELECT privilege is required in addition to UPDATE or DELETE privilege if the standards option specifies:

- The SQLRULES(STD) bind option for static SQL statements
- The CURRENT RULES special register is set to STD for dynamic SQL statements.

When the SELECT privilege is required, but the user is not authorized, SQLCODE '-551' is returned with SQLSTATE '42501'.

## REFERENCES Privilege on Columns

REFERENCES privilege is required to define referential constraints. GRANT is extended to support a list of columns for the REFERENCES privilege. However, the REVOKE statement is not extended; therefore, if you specify REFERENCES in a REVOKE statement the privilege is revoked from all columns that were previously granted by the grantor.

---

## Migration Step 25: Convert Catalog and Directory Indexes (Optional)

This step uses the CONVERT option of the CATMAINT utility to convert type 1 indexes on the DB2 catalog and directory to type 2 for improved concurrency. You can run this job to convert your indexes before or after migration. Skip this step if you do not have type 1 indexes or if you do not want to convert your type 1 indexes. The DISPLAY UTILITY command cannot provide information about CATMAINT status while the job is running.

If you are enabling data sharing, you must run job DSNTIJCX first to convert indexes in the catalog, directory, communication database, DDL registration database, and resource limit facility to type 2.

You can modify the job to convert these indexes back to type 1. See *SQL Reference* for a complete listing of the catalog indexes. See Section 2 of *Utility Guide and Reference* for information about the CATMAINT utility. You can also convert indexes to type 2 in these ways:

- Use the CONVERT TO TYPE 2 option of the ALTER INDEX SQL statement to convert catalog indexes. This method works only with catalog indexes, not directory indexes.
  1. Be aware that Type 2 indexes require more space than Type 1 indexes. Before you convert to Type 2 indexes with the ALTER INDEX command, determine if you need to allocate more space for these indexes. Refer to “Calculating the Space Required for an Index” on page 2-29 for more information. Based on these calculations, if your index space is not large enough, delete and re-define your data sets with a larger allocation. Then run the utility RECOVER INDEX. This RECOVER formats the empty page. See the *Utility Guide and Reference* for more information on RECOVER INDEX. You can use DFSMSdftp to manage your data sets. For more information on increasing your data set allocations, see *DFSMS/MVS: Storage Administration Reference for DFSMSdftp*.
  2. Issue ALTER INDEX *index-name* CONVERT TO TYPE 2 for each index.
  3. You must run the RECOVER INDEX utility for each index. This RECOVER formats to the altered index type.
- Use the CONVERT TO TYPE 2 option of the CATMAINT utility to convert catalog and directory indexes.
  1. Be aware that Type 2 indexes require more space than Type 1 indexes. Before you convert to Type 2 indexes with the CATMAINT utility, determine if you need to allocate more space for these indexes. Refer to “Calculating the Space Required for an Index” on page 2-29 for more information. Based on these calculations, if your index space is not large enough, delete and re-define your data sets with a larger allocation. Then run the utility RECOVER INDEX. This RECOVER formats the empty page. See the *Utility*







---

## Chapter 2-8. Falling Back and Remigrating

Falling back is the process of returning to DB2 for MVS/ESA Version 4 after migrating to DB2 for OS/390 Version 5. Fall back if you have a severe error while operating Version 5 and you want to return to operation on Version 4.

Remigrating is the process of returning to Version 5 after falling back to Version 4.

---

### Falling Back

Because the structure of the DB2 for OS/390 Version 5 catalog is used in the DB2 for MVS/ESA Version 4 after falling back, the fallback procedure involves only a few steps:

1. Run Phase 0 of the Version 5 installation verification procedure
2. Stop Version 5 activity
3. Reactivate Version 4
4. Reconnect TSO, IMS, and CICS to Version 4
5. Start Version 4
6. Verify fallback.

You can save your Version 5 TSO LOGON procedures and JCL for remigration to Version 5.

### Fallback Considerations

To avoid complications, do not use the new DB2 for OS/390 Version 5 facilities until you are certain that you will not need to fall back.

#### Recovery Restrictions

Image copies that are created with the COPYDDN, RECOVERYDDN, or SHRLEVEL REFERENCE keywords of REORG or LOAD REPLACE cannot be used for recovery on a Version 4 system. The RECOVER utility must run on a Version 5 system to establish a recoverable point for these copies. Consider making full image copies with the COPY utility *before* or *after* falling back so that a recoverable point is available.

#### Automatic Rebind

After fallback, plans or packages bound in Version 5 will be automatically rebound on first execution in Version 4. See chapter 4 of *Application Programming and SQL Guide* for more details on automatic rebinding.

#### Data Sharing

There are additional considerations for falling back if any member of a data sharing group falls back. See *Data Sharing: Planning and Administration* for these details.

#### Frozen Objects

Falling back does not undo changes made to the catalog during migration to Version 5. The migrated catalog is used after fallback. Some objects in this catalog that have been affected by Version 5 function might become *frozen* objects after fallback. Frozen objects are unavailable, and they are marked with the release dependency marker H. If an object is marked with a release dependency, it is never

unmarked. Using the ALTER TABLE statement to remove the dependency marker does not work.

In general, objects that depend on the new facilities of DB2 for OS/390 Version 5 are frozen after you fall back to DB2 for MVS/ESA Version 4 until you remigrate to Version 5. The objects listed in Table 67 are frozen.

Table 67. Objects Frozen When Falling Back to DB2 for MVS/ESA Version 4

| Release<br>Dependent<br>Mark | Frozen Objects                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| H                            | <ul style="list-style-type: none"> <li>• Plans that use any new syntax or objects</li> <li>• Packages that use any new syntax or objects</li> <li>• Plans and packages that refer to ASCII tables</li> <li>• Temporary tables</li> <li>• Views defined with WITH CHECK OPTION</li> <li>• Tables having a foreign key that refers to a unique non-primary key</li> <li>• Tables having a foreign key with a delete action of NO ACTION</li> <li>• Tables having a foreign key with referential constraints that refer to a parent key which is a unique non-primary key (child tables, parent tables, and constraint indexes)</li> <li>• Tables, indexes, or views that are 255 bytes long.</li> <li>• Unique indexes on tables if the index satisfies an incomplete constraint for a foreign key that refers to a unique non-primary key (UNIQUERULE is set to R)</li> <li>• Large table spaces and the tables they contain</li> <li>• Table spaces created with keyword MEMBER CLUSTER, and the tables these table spaces contain and the indexes that reference those tables</li> <li>• Image copies produced with the COPYDDN or RECOVERYDDN keywords of REORG or LOAD REPLACE</li> <li>• ASCII tables, table spaces, databases, plans, and packages</li> <li>• Table spaces altered to selective partition locking</li> <li>• CDB tables in the catalog not used in DB2 for MVS/ESA Version 4 (You must use your Version 4 CDB.)</li> </ul> |

**Plans and packages** do not become frozen objects because of new SQL syntax. If you try to use plans or packages that contain new SQL syntax, the automatic rebound that takes place the first time you try to run the plan in the fallback release fails. To make the plans and packages that were not automatically rebound available, change the offending SQL statements, precompile the application programs, and bind the plans and packages.

General-use Programming Interface

While operating in Version 4, you can determine if any of your objects are frozen by issuing the following SELECT statements:

```

SELECT NAME FROM SYSIBM.SYSPLAN
WHERE IBMREQD = 'H';
SELECT LOCATION, COLLID, NAME, VERSION FROM SYSIBM.SYSPACKAGE
WHERE IBMREQD = 'H';
SELECT CREATOR, NAME FROM SYSIBM.SYSVIEWS
WHERE IBMREQD = 'H';
SELECT CREATOR, NAME FROM SYSIBM.SYSINDEXES
WHERE IBMREQD = 'H';
SELECT CREATOR, NAME, TYPE FROM SYSIBM.SYSTABLES
WHERE IBMREQD = 'H';
SELECT DBNAME, NAME FROM SYSIBM.SYSTABLESPACE
WHERE IBMREQD = 'H';

```

\_\_\_\_\_ End of General-use Programming Interface \_\_\_\_\_

## Other Fallback Considerations

After you have fallen back to Version 4 and you are operating on that subsystem, you should be aware of the following operational considerations:

**Stored procedures:** SQLCODE -417 is given for stored procedures from a Version 4 subsystem that access the Version 5 SYSPROCEDURES catalog table. The table has new columns that are not compatible with stored procedures requests if any of these conditions exist:

- RESULTS\_SETS is a non-zero number
- WLM\_ENV is non-blank
- PGM\_TYPE is a value other than 'M'
- EXTERNAL\_SECURITY is a value other than 'N'
- COMMIT\_ON\_RETURN is 'Y'

**Access control authorization exit:** If you have used DSNX@XAC to bypass some or all of the DB2 authorization checking, then the DB2 catalog has no GRANTS in the authorization tables. It is possible to build and execute GRANT statements from the authorization exit to prevent fallback problems. For more information on the access control authorization exit, see Appendix B (Volume 2) of *Administration Guide*.

**MAXROWS:** The value specified for MAXROWS is not honored in Version 4, although the MAXROWS column of SYSIBM.SYSTABLESPACE receives a value of 255.

**Values of Partitioning Keys:** After falling back to Version 4, you cannot update a value that is part of a partitioning key.

**Nonpartitioned indexes:** A nonpartitioned index created in Version 5 with a non-default or altered PIECESIZE cannot be accessed after falling back to Version 4. You can drop the index before falling back, or use the REPAIR utility (REPAIR DBD DROP INDEX) to drop the index while running in Version 4.

**Table check constraints:** Table check constraints are added to the catalog during migration and are not removed after falling back. Therefore, some catalog columns might be set to -1. You might want to execute queries similar to the following to see what changed during migration:

\_\_\_\_\_ General-use Programming Interface \_\_\_\_\_

```

SELECT TBNAME,FREQUENCY,FREQUENCYF,
NAME,CARDF,NUMCOLUMNS
FROM SYSIBM.SYSCOLDIST
WHERE CARDF < -1 OR NUMCOLUMNS < 1 OR
 FREQUENCY < 0 OR FREQUENCY > 10000 OR
 FREQUENCYF < 0 OR FREQUENCYF > 1;

```

\_\_\_\_\_ End of General-use Programming Interface \_\_\_\_\_

**Catalog changes:** There are new columns in catalog tables SYSCOLDIST and SYSCOLDISTSTATS, and some columns in those tables that have new definitions. The new columns are TYPE, CARDF, NUMCOLUMNS and COLGROUPOCOLNO. These columns contain statistics that are collected on concatenated key columns of an index, instead of just the first key column of an index. After falling back, it is possible that a Version 4 subsystem might be using a Version 5 catalog. In this case, DB2 could see statistics for concatenated columns and interpret them as single column statistics.

## Step 1: Run Phase 0 of the DB2 for OS/390 Version 5 Installation Verification Procedure

This step removes all the installation verification processing done on DB2 for OS/390 Version 5. When you run this step, all the DB2 for OS/390 Version 5 sample tables and plans are deleted. You re-create these objects when you remigrate to Version 5.

## Step 2: Stop DB2 for OS/390 Version 5 Activity

Ensure that no recovery is required on system databases.

To stop Version 5 work, perform the following steps:

1. Issue the command:

```
-DSN1 STOP DB2 MODE(QUIESCE)
```

The QUIESCE keyword allows DB2 to complete processing of currently executing programs. This might require some processing time.

2. Issue the command:

```
-DSN1 START DB2 ACCESS(MAINT)
```

This allows only the install-defined system administrators and system operators to access DB2.

If DB2 does not start properly, it usually abends with a reason code indicating where the error occurred. To find the error, check the set of definitions for the associated resource. For example, if the BSDS does not match the subsystem parameter values, check to see that the correct jobs were run for DSNTIJUZ. Check to see that you started DB2 with the correct subsystem parameter.

3. Make sure all work is complete.

- Make sure no units of recovery remain. Issue the command:

```
-DSN1 DISPLAY THREAD(*) TYPE(*)
```

Then use -RECOVER INDOUBT for any indoubt threads.

- Make sure no utility work remains. Issue the command:

-DSN1 DISPLAY UTILITY(\*)

Then, either allow utilities to complete before proceeding, or stop all utility processing with the command:

-DSN1 TERM UTILITY(\*)

- Make sure no table spaces and index spaces in the DB2 directory (DSNDB01) or the DB2 catalog (DSNDB06) have write error ranges or deferred restart states. Issue the command:

-DSN1 DISPLAY DATABASE(DSNDB01) SPACENAM(\*) RESTRICT

-DSN1 DISPLAY DATABASE(DSNDB06) SPACENAM(\*) RESTRICT

A user with install-defined system administrator or system operator authority also must enter this command.

Recover any table spaces and index spaces with write error range or deferred restart states.

4. To stop DB2, issue the command:

-DSN1 STOP DB2 MODE(QUIESCE)

A user with install-defined system administrator or system operator authority also must enter this command.

If IRLM does not stop automatically when DB2 stops, stop IRLM manually. To stop IRLM, issue the command:

STOP *irlmproc*

where *irlmproc* is the name you assigned to the IRLM startup procedure.

### Step 3: Reactivate DB2 for MVS/ESA Version 4 Code: DSNTIJFV

This job renames procedures to activate DB2 for MVS/ESA Version 4 and deactivate DB2 for OS/390 Version 5. It defaults to SYS1.PROCLIB as the target for JCL procedures. Add statements to rename other procedures as well, such as your IMS, CICS, TSO logon, and batch procedures. You might also need to rename procedures in your jobs from Version 4.

You might want two sets of procedures, such as DSN1xxxx and DSN2xxxx, at all times, with an alias for the current release level.

If DSNTIJFV runs successfully, it produces a return code of 0. Check to make sure all renames execute successfully.

If DSNTIJFV fails or abends, rerun only the renames that failed. If some of the procedures already exist, check carefully to ensure that procedures for the two releases are not mixed.

### Step 4: Reconnect TSO, IMS, and CICS to DB2 for MVS/ESA Version 4

Reestablish your Version 4 logon procedures and JCL, as well as the CICS Version 4 and IMS connections.

If you overwrote the load module during migration to DB2 for OS/390 Version 5, reassemble the RCT with the Version 4 libraries.

If you did not overwrite the load module, change the STEPLIB statements for CICS and IMS jobs so they refer to the Version 4 libraries.

## Step 5: Start DB2 for MVS/ESA Version 4

Perform the following steps to start Version 4:

1. Start the IRLM.

If you have not requested that DB2 automatically start the IRLM, start it before you start DB2. Use the command:

```
START irlmproc
```

where *irlmproc* is the name you assigned to the IRLM startup procedure. This is the value you specified for the PROC NAME option on installation panel DSNTIPI.

If you specified YES for the AUTO START option on installation panel DSNTIPI, DB2 starts the IRLM automatically.

2. Start DB2 from the MVS console using the command:

```
-DSN1 START DB2,PARM(DSNZPxxx)
```

where *-DSN1* is the subsystem command prefix you defined for DB2, and *DSNZPxxx* is the name of the Version 4 subsystem parameter module. If you used the default name, DSNZPARM, you can omit the PARM parameter.

If DB2 starts successfully, 2 to 5 address spaces also start. These address spaces are *ssnmMSTR* and *ssnmDBM1*, and possibly *ssnmSPAS*, *ssnmDIST*, and *irlmproc*, where *ssnm* is the DB2 subsystem name and *irlmproc* is the IRLM procedure name.

If DB2 starts successfully, the series of restart messages you receive concludes with these two messages:

```
DSNR002I RESTART COMPLETED
DSN9022I DSNYASCP '-DSN1 START DB2' NORMAL COMPLETION
```

3. **If you have done distributed processing with your DB2 for OS/390 Version 5 subsystem**, check message DSNR005I for the number of INDOUBT threads after you start DB2.

If there are no INDOUBT threads, continue falling back as if you had not done any distributed processing. If there are INDOUBT threads, issue the command:

```
-DSN1 DISPLAY THREAD(*) TYPE(INDOUBT)
```

If the number of INDOUBT threads reported in the DSNV408I messages is equal to the number of threads reported in the DSNR005I message, continue falling back as if you had not done any distributed processing. If there are fewer INDOUBT threads reported by DSNV408I messages than in message DSNR005I, proceed as follows:

- a. Stop DB2 for MVS/ESA Version 4.
- b. Determine which units of work are incomplete by scanning the DB2 recovery log with the DB2 for OS/390 Version 5 DSN1LOGP utility. Use the SUMMARY option of this utility.
- c. Examine the DSN1LOGP output to find all the DSN1162I messages that have a COORDINATOR name in a remote location. Each of these messages identify an INDOUBT DBAT. Record the LUWID displayed in each message.

- d. Decide whether to COMMIT or ABORT each INDOUBT DBAT. One way to do this is by contacting the COORDINATOR location. If it is another DB2, use the DISPLAY THREAD command to help you decide.
  - e. If you have not already done so during migration, apply the fallback PTF supplied with Version 5.
  - f. Start Version 4 again.
  - g. Issue the RECOVER INDOUBT ACTION(*correct decision*) LUWID(*luwid*) command to resolve each INDOUBT DBAT.
4. **If you have not done distributed processing with your DB2 for OS/390 Version 5 subsystem**, check outstanding restrictions after you start DB2. Identify databases whose uses are restricted with the command:
- ```
-DSN1 DISPLAY DATABASE(*) SPACENAM(*) RESTRICT
```
- You can start some of these databases at this time. For more information on starting restricted databases, see Section 4 (Volume 1) of *Administration Guide*.
5. If DB2 does not start properly, it usually abends with a reason code indicating where the error occurred. To find the error, check the set of definitions for the associated resource. A common cause of startup failure is that the BSDS does not match the subsystem parameter values; be sure that the startup procedure is pointing to the correct BSDS and subsystem parameter. Also, check that the subsystem parameter member you specified (or allowed to default) when you started DB2 is the one built by job DSNTIJUZ. Check the JCL for the DB2 startup procedure.
 6. Optionally, start TSO. If you want to use the TSO SUBMIT command to do housekeeping and fallback verification, you must start TSO (if it is not already started).

Step 6: Verify Fallback

At this point, you must perform some of your own testing to determine if the fallback was successful. You cannot run the DB2 for OS/390 Version 5 samples on DB2 for MVS/ESA Version 4.

How to Verify Fallback:

1. Run the Version 4 sample applications
2. Test your own applications
3. Retry the problem for which you decided to fall back.

Be aware that there are some operational considerations when operating on Version 4. These are described in “Other Fallback Considerations” on page 2-299.

Remigrating

Migrating after falling back (remigrating) is similar to the normal migrating process.

When remigrating, refer to “Migration Considerations” on page 2-267 because many of those considerations apply to remigrations, too. Which considerations apply depends on the type of activity that took place on your DB2 for MVS/ESA Version 4 subsystem after falling back.

Make sure your DB2 for OS/390 Version 5 CDB and Version 4 CDB are both current when you remigrate.

Several updatable catalog columns are copied to new columns during migration. The columns are not recopied during remigration, so update the renamed columns before remigrating. See “Adjust Application Programs” on page 2-290 for the copied columns.

A plan or package is automatically rebound in Version 5 when executed for the first time after remigration if it was not explicitly bound in Version 4. However, if you specified NO for the AUTO BIND option on installation panel DSNTIPO, then automatic binds are disabled. This means that the plan or package from your previous release is the one that runs in Version 5. This means the plan or package does not benefit from Version 5 enhancements.

When remigrating, you do **not** have to:

- Allocate the target and distribution libraries
- Run the SMP/E jobs
- Run the installation CLIST
- IPL MVS.

When remigrating, you **do** have to:

1. Run DSN1COPY with the CHECK option on the Version 4 catalog table spaces. Also, run DSN1CHKR on Version 4. For information on these utilities, see “Migration Step 1: Run Link Checker on DB2 for MVS/ESA Version 4 Table Spaces (optional)” on page 2-272. Finally, execute the queries in member DSNTESQ of *prefix.SDSNSAMP*.

2. Image copy the Version 4 catalog using the DSNTIJIC job.

Although this step is not required, it is recommended. See “Migration Step 4: Image Copy Directory and Catalog in Case of Fallback: DSNTIJIC” on page 2-274 for more information.

3. Stop DB2 for MVS/ESA Version 4.

4. Reconnect TSO, IMS, and CICS to DB2 for OS/390 Version 5.

Reestablish your Version 5 logon procedures and JCL, as well as your Version 5 CICS and IMS connections.

5. Rebuild Version 5 cataloged procedures.

Rename the Version 5 procedures that were renamed by job DSNTIJFV during fallback. If job DSNTIJFV was not run, job DSNTIJMV must be re-run. Comment out step 1 (DSNTIMP), which defines Version 5 to MVS, and run the job. (There is no need to define Version 5 to MVS a second time.)

6. Start DB2 for OS/390 Version 5.

Make sure you are using your Version 5 subsystem parameter.

7. Image copy the Version 5 catalog using the DSNTIJIC job.

For information on job DSNTIJIC, see “Migration Step 4: Image Copy Directory and Catalog in Case of Fallback: DSNTIJIC” on page 2-274.

8. Verify your DB2 for OS/390 Version 5 system.

For information on this procedure, see “Migration Step 23: Verify Your DB2 for OS/390 Version 5 System” on page 2-289.

Chapter 2-9. Verifying with the Sample Applications

Use the sample applications to verify either installation or migration. During verification, run all sample applications under the same user ID; this user ID must have SYSADM authority. Otherwise, errors may occur.

If you are migrating, we recommend that you run portions of the sample applications from DB2 for MVS/ESA Version 4 after you finish migration. This verifies the migration and ensures that the old jobs work with DB2 for OS/390 Version 5. Version 5 sample applications can be run next. For information on how to run the sample applications from the previous release, see “Migration Step 23: Verify Your DB2 for OS/390 Version 5 System” on page 2-289.

The installation verification procedure consists of seven phases: six verification phases and one cleanup phase that drops sample objects. Each of the six verification phases tests one or more DB2 functions or attachment facilities. Certain phases of the verification procedure might not apply to the environment in which your DB2 subsystem operates, so you might not perform all phases. In some cases, the steps and return codes differ when running the fallback release and Version 5 phases. These differences are noted under the proper phase.

To help you perform the verification procedure, DB2 provides several jobs that invoke sample applications. You run the same jobs whether you are installing DB2 for the first time or migrating from your Version 4.

These jobs have been tailored and loaded by the DSNTINST CLIST into *prefix*.NEW.SDSNSAMP that you created during your installation or migration. Sometimes the verification jobs access information from the untailored *prefix*.SDSNSAMP library that existed before installation or migration.

If you are installing a data sharing group, run the installation verification procedure (IVP) after you install or migrate the originating system. You do not need to run the IVP after you enable the originating system or after you install a new data sharing member. See the installation procedures in *Data Sharing: Planning and Administration* for more information about verifying that your data sharing definitions are correctly established and that Sysplex parallelism is enabled.

Do not delete the fallback release sample objects from your subsystem. You need them to verify the success of a migration to Version 5 and to fall back to DB2 for MVS/ESA Version 4 in case of a Version 5 failure. If you are migrating, you can run the fallback release sample programs on Version 5 without any preparation. This provides a test of the migration process.

Most DB2 sample objects have unique names to differentiate them from objects of previous releases. This allows sample programs for multiple releases to coexist. There is an exception—a table called PLAN_TABLE—which is created in Phase 3 of the verification procedure for the EXPLAIN statement. The name of this table does not change from Version 4 to Version 5.

The JCL provided for CICS and IMS sets up transaction identifiers for the sample applications.

Installation Verification Phases and Programs

Table 68 shows the programs that are run in each phase of the verification procedure. These programs need to be run sequentially by phase because the output of some jobs is used as input for following jobs. You must use the same compiler for each job. For example, if you use COB2 for DSNTEJ2C, you must use COB2 for all COBOL verification programs.

Run Phase 0 (job DSNTEJ0) only if you want to remove all the verification processing you have completed so that you can begin the verification procedure again. Phases 1-3 test the TSO and batch environments. Phase 4 is for IMS users only, and Phase 5 is for CICS users only. Phase 6 sets up the sample tables for distributed processing.

prefix.SDSNSAMP contains the program source. Details on how to print it are provided in “Printing the Sample Application Listings” on page 2-341.

When you complete the verification procedure, save the verification objects; you need them when you migrate to the next release of DB2.

The jobs listed in Table 68 are designed to run with minimal interaction on your part. However, before running these jobs, make any modifications suggested either in this chapter or in “Completing the CLIST Processing” on page 2-219.

After running the verification jobs, you can still fall back to DB2 for MVS/ESA Version 4. See “Falling Back” on page 2-297 for details.

Table 68 (Page 1 of 3). Relationship of Phases to Programs

Phase	Job	Program	Description
0	DSNTEJ0	DSNTIAD	Remove sample applications
1	DSNTEJ1	DSNTIAD	Create tables
		DSN8CA	Assembler interface to call attach facility
		DSN8EAE1	Edit exit
		DSN8HUFF	Huffman compression exit
		DSNUTILB	Utilities
	DSNTEJ1P	DSNTEP2	Dynamic SQL program
	DSNTEJ1S ¹	DSNHSP	
	DSNTEJ1T ²	DSNUPROC	

Table 68 (Page 2 of 3). Relationship of Phases to Programs

Phase	Job	Program	Description
2	DSNTEJ2A	DSNTIAD	Grant execution
		DSNTIAUL	Unload and load tables
		DSNUTILB	Utilities
	DSNTEJ2C	DCLGEN	Generate declarations
		DSNTIAD	Grant execution
	DSNTEJ2D	DSN8BC3	COBOL phone application
		DSNTIAD	Grant execution
	DSNTEJ2E	DSN8BD3	C phone application
		DSN8MDG	Prepare error message routine
	DSNTEJ2F	DSN8BECL	Prepare classes used by C++ phone application
		DSNTIAD	Grant execution
		DSN8BE3	C++ phone application Grant execution
		DSNTIAD	FORTTRAN phone application
DSN8BF3		Generate declarations	
DSNTEJ2P	DCLGEN	Grant execution	
	DSNTIAD	PL/I phone application	
DSN8BP3			
3	DSNTEJ3C	SPUFI	Sample SPUFI input
		DSNTIAD	Grant execution
		DSN8CC	COBOL interface to call attachment facility
	DSNTEJ3P	DSN8SCM	COBOL connection manager
		DSN8SC3	COBOL phone application
		DSN8HC3	COBOL organization application
		DSNTEP2	Dynamic SQL application
		DSNTIAD	Grant execution
		DSN8SPM	PL/I connection manager
		DSN8SP3	PL/I phone application
4	DSNTEJ4C	DSNTIAD	Grant execution
		DSN8ICx	Organization application
	DSNTEJ4P	DSN8MCx	Copy code
		DSNTIAD	Grant execution
		DSN8IPx	Organization, project applications
DSN8MPx	Copy code		
5	DSNTEJ5A	DSNTIAC	CICS SQLCA formatter front-end
		DSNTIAD	Grant execution
	DSNTEJ5C	DSN8CCx	Organization application
		DSN8MCx	Copy code
		DSNTIAD	Grant execution
		DSN8CPx	Organization, project applications
DSN8MPx	Copy code		
6	DSNTEJ6	DSNTIAD	Update location column in the department table to the sample location entered at installation time.
	DSNTEJ6P ³	DSN8EP1	Invoke the sample stored procedure.
	DSNTEJ6S ³	DSN8EP2	Create sample stored procedure.
	DSNTEJ6D ³	DSN8ED1	Compile, link-edit, bind and run stored procedure sample application
	DSNTEJ6T ³	DSN8ED2	Register, prepare, and bind the stored procedure sample application
	DSNTEJ61	DSN8EC1	Prepare sample ODBA stored procedure
	DSNTEJ62	DSN8EC2	Call ODBA stored procedure

Table 68 (Page 3 of 3). Relationship of Phases to Programs

Phase	Job	Program	Description
Note:			
			1. Job DSNTJ1S, which contains the sample JCL to run the schema processor, is not a part of the sample applications to verify installation. For more information on the schema processor, see Section 2 (Volume 1) of <i>Administration Guide</i>
			2. Job DSNTJ1T, used for adding rows to SYSIBM.SYSSTRINGS for character conversion purposes, is not a part of the sample applications to verify installation.
			3. The Phase 6 sample jobs, DSNTJ6T, DSNTJ6D, DSNTJ6S, and DSNTJ6P are not edited by the CLIST unless a REMOTE LOCATION name is entered on panel DSNTIPY, a LOCATION NAME is entered on panel DSNTIPR, and you specify PL/I for MVS and VM on panel DSNTIPG.

|
|

#

Planning for Verification

Before performing any of the verification phases, you must make certain decisions about your verification strategy. DB2 system administrators and system administrators for ISPF, TSO, batch, IMS, and CICS must be involved in these decisions. With these system administrators:

- Determine the verification phases you plan to perform.
Examine the description of each verification phase in this chapter, and determine which phases apply to your needs.
- Identify any phases you want to modify before you perform them.
Verification is designed to run with little interaction on your part. This chapter does not discuss how to modify any of the phases, but you can adapt any of the seven phases to your needs. If this is your intent, identify and describe any modifications you plan to make.
- Establish additional testing steps to complete the verification.
The six verification phases and the jobs you run to perform them are valuable tools for testing DB2. They are not a substitute for a thorough subsystem test. You must plan and perform your own additional testing to complete the verification. To help you assess which additional tests might be necessary, examine the sample applications provided with DB2.
- Start any DB2 databases that are not currently started.

Special Considerations for COBOL Programs

The DB2 COBOL samples were tested with the following compiler options. If you have a problem executing the DB2 COBOL samples, ensure that your compiler options are consistent with the IBM COBOL options in Table 69 on page 2-309 or the COBOL options in Table 71 on page 2-310 or the COB2 options in Table 70 on page 2-309. Remember that if you are using CICS, the options you need to use depend on the CICS environment. To verify that you are using the correct options in your CICS environment, refer to *CICS/ESA Application Programming Guide*.

|

Table 69. IBM COBOL Options (formerly Cobol/370)

ADV	NOEXIT	NOTYPECHK
BUFSIZE(4096)	NOFASTSRT	NOVBREF
DATA(31)	NOFLAGMIG	NOWORD
FLAG(I)	NOFLAGSTD	NOXREF
INTDATE(ANSI)	NOIDLGEN	NUMPROC(NOPFD)
LANGUAGE(EN)	NOLIB	OBJECT
LINECOUNT(60)	NOLIST	OUTDD(SYSOUT)
NOADATA	NOMAP	PGMNAME(LONGUPPER)
NOAWO	NONAME	QUOTE
NOCMPR2	NONUMBER	RENT ³
NOCOMPILE(S)	NOOFFSET	RMODE(AUTO)
NOCURRENCY	NOOPTIMIZE	SIZE(MAX)
NODBCS	NOSEQUENCE	SOURCE
NODECK	NOSSRANGE	SPACE(1)
NODUMP	NOTERM ¹ or TERM ²	TRUNC(STD)
NODYNAM	NOTEST	ZWB

Notes:

¹ Refers to jobs DSNTEJ2C, DSNTEJ3C, and DSNTEJ4C only

² Refers to job DSNTEJ5C only

³ See the CICS documentation for actual options to use

Table 70. COB2 (VS COBOL II) Options

ADV	NOMAP
APOST	NONAME
BUFSIZE(4096)	NONUMBER
DATA(31)	NOOFFSET
FLAG(I)	NOOPTIMIZE
LANGUAGE(EN)	NOSOURCE ¹ or SOURCE ²
LIB ¹ or NOLIB ²	NOSSRANGE
LINECOUNT(60)	NOTERM
NOAWO	NOTEST
NOCMPR2	NOVBREF
NOCOMPILE(S)	NOWORD
NODBCS	NOXREF ¹ or XREF ²
NODECK	NUMPROC(NOPFD)
NODUMP	OBJECT
NODYNAM	OUTDD(SYSOUT)
NOEXIT	RENT ³
NOFASTSRT	RESIDENT ³
NOFDUMP	SEQUENCE
NOFLAGMIG	SIZE(MAX)
NOFLAGSA	SPACE(1)
NOFLAGSTD	TRUNC(OPT)
NOLIST	ZWB

Notes:

¹ Refers to job DSNTEJ5C only

² Refers to jobs DSNTEJ2C, DSNTEJ3C, and DSNTEJ4C only

³ See the CICS documentation for actual options to use

Table 71. COBOL Options

ADV	NOCOUNT	NOSTATE
APOST	NODECK	NOSUPMAP
BUF=12288 ¹	NODMAP	NOSYMDMP
BUF=64000 ²	NODYNAM	NOSYNTAX
COMPILE=01	NOENDJOB	NOTERM
DUMP	NOFDECK	NOTEST
FLAGW	NOFLOW	NOTRUNC
LANGLVL(2)	NOLIB	NOVBREF
LCOL2	NOLST ¹	NOVBSUM
LINECNT=57	NOLVL	SEQ
LOAD	NOMIGR	SOURCE ¹
LSTCOMP ²	NONAME	or NOSOURCE ²
L120 ¹	NONUM	SPACE1
L132 ²	NOOPTIMIZE	SXREF ³
NOBATCH	NOPMAP	or NOSXREF ⁴
NOCDECK	NOPRINT	SYST
NOCLIST	NORESIDEN ⁷	VERB
		XREF ⁵
		or NOXREF ⁶
		ZWB

Notes:

- 1 Refers to jobs DSNTJ2C, DSNTJ3C, and DSNTJ4C only
- 2 Refers to job DSNTJ5C only
- 3 Refers to jobs DSNTJ2C and DSNTJ4C only
- 4 Refers to jobs DSNTJ3C and DSNTJ5C only
- 5 Refers to job DSNTJ3C only
- 6 Refers to jobs DSNTJ2C, DSNTJ4C, and DSNTJ5C only
- 7 See the CICS documentation for actual options to use

COBOL programs can use the VS COBOL II (COB2) compiler or the IBM COBOL for MVS & VM (formerly IBM SAA AD/Cycle COBOL/370 (COBOL/370)) compiler and even OS/VS COBOL (though we recommend that you use VS COBOL II or IBM COBOL for MVS & VM).

If you wish to run your samples applications with IBM COBOL or COB2, but you selected an earlier version of COBOL on install panel DSNTIPF, or you just want to test with an additional version of COBOL, you can do the following to get updated installation verification procedures (IVP):

- Run the installation CLIST in **INSTALL** mode
- On field INPUT MEMBER NAME (field 6) in panel DSNTIPA1, use the name of the defaults file in which the defaults for your existing DB2 are stored
- View all other installation panels and use default values **with the following exceptions:**
 - Change the data set names in fields 1 and 3 on panel DSNTIPT. This prevents the *prefix.NEW.DSNSAMP* and *prefix.NEW.DSNTTEMP* data sets from your original installation from being overwritten.
 - On panel DSNTIPQ, make sure that you have entered the correct data set names for the type of COBOL for which you want new IVP jobs.
 - On panel DSNTIPY, change the value in field COBOL TYPE (field 2) to the type of COBOL for which you want new IVP jobs.
- When the installation CLIST completes, the new *prefix.NEW.DSNSAMP* data set that you specified on panel DSNTIPT contains the updated IVP jobs

If you would like information on the language CLIST (DSNH), refer to the *Command Reference*.

Notes for users of IBM SAA AD/Cycle COBOL/370 (COBOL/370) Version 1 Release 1

- Change the VS COBOL II data set names in the DSNHCOB2 JCL procedure to the corresponding COBOL/370 names. This is done automatically if you enter the COBOL/370 data set names in the installation CLIST panel DSNTIPU.
- Increase the region size to 2MB for jobs DSNTEJ2C and DSNTEJ5C for the job to complete successfully. Other jobs can use the JES2 default region size.

For more detailed instructions, see *IBM COBOL for MVS & VM Programming Guide* and *Language Environment for MVS & VM Programming Guide*.

Special Considerations for C and C++ Programs

The DB2 C and C++ samples were tested with the following compiler options. If you have a problem executing the DB2 C and C++ samples, ensure that your compiler options are consistent with the options in table Table 72 or Table 73.

Table 72. C Language Options

NOAGGR	NOEVENTS	NOHWOPTS	NOMEMORY	NOPPONLY
NOALIAS	EXECOPS	NOINLINE ⁴	NESTINC(255)	REDIR ¹
ARGPARSE ¹	NOEXPMAC	LIST	OBJECT	NORENT
NOCHECKOUT ²	NOEXPORTAL ³	NOLOCALE	NOOE ¹	NOSEARCH
NOCSECT	FLAG(I)	NOLONGNAME	NOOFFSET	NOSHOWINC
NODECK	NOGONUMBER	NOLSEARCH	NOOPTIMIZE	SOURCE
NODLL ³	HALT(16) ¹	MAXMEM(2000) ⁵	PLIST(HOST) ¹	SPILL(128) ⁵
NOSSCOMM	START	TARGET(LE)	TERMINAL	NOTEST ⁶
NOUPCONV	XREF			

1. This option is used only by IBM C/C++ for MVS/ESA V3R2
2. NOPPTRACE, PPCHECK, GOTO, ACCURACY, PARM, NOENUM, NOEXTERN, TRUNC, INIT, NOPORT, GENERAL
3. This option is used only by IBM C/C++ for MVS/ESA V3R1 and V3R2
4. AUTO, NOREPORT, 100, 1000
5. This option is used only by IBM AD/Cycle C/370 V1R2
6. SYM, BLOCK, LINE, NOPATH

Table 73. C++ Language Options

ARGPARSE	NOIDL ¹	NESTINC(255) ²	NOSHOWINC	TERMINAL
NOATTRIBUTE	NOINFO	OBJECT	NOSOM	NOTEST
NOCSECT	NOINLRPT	NOOE ²	SOMEINIT	XREF
EXECOPS	LANGLVL(EXTENDED)	NOOFFSET	NOSOMGS	HALT(16)
NOEXPMAC	NOLIST	OPTIMIZE(0)	SOURCE	MEMORY ²
NOEXPORTALL	NOLOCALE ²	PLIST(HOST) ²	NOSRCMSG	NOSEQUENCE
FLAG(I)	LONGNAME	NOPPONLY	START ²	TEMPINC
NOGONUMBER	MARGINS	REDIR	TARGET(LE) ²	

1. This option is used only by IBM C/C++ for MVS/ESA V3R1.
2. This option is used only by IBM C/C++ for MVS/ESA V3R2.

The installation CLIST customizes C++ compiler parameters in sample job DSNTEJ2E if you have specified C++ for MVS/ESA V3R2 on panel DSNTIPU.

If you wish to run your samples applications with C++ for MVS/ESA V3R2, but you selected an earlier version of that language on the installation panel, you should make the following changes:

- In sample job DSNTEJ2E, change all occurrences of

```

    PARM.CP='SOURCE XREF,MARGINS',
    to
    PARM.CP='/CXX SOURCE XREF,MARGINS',
    • In sample procedure DSNHCPP2, change
    //CP      EXEC PGR=CBC320PP,COND=((4,LT,PC1),(4,LT,PC2)),
    //        REGION=4096K
    to
    //CP      EXEC PGR=CBC320PP,COND=((4,LT,PC1),(4,LT,PC2)),
    //        REGION=4096K,PARM='/CXX'

```

Special Considerations for PL/I Programs

The DB2 PL/I samples were tested with the compiler options shown in Table 74. If you have a problem executing the DB2 PL/I samples, ensure that your compiler options are consistent with these options.

Table 74. PL/I Options

CHARSET(60,EBCDIC)	LINECOUNT(55)	LMESSAGE	MARGINS(2,72,0)	NOAGGREGATE
NOATTRIBUTES	NOCOMPILE(S)	NOCOUNT	NOESD	NOFLOW
NOGONUMBER	NOGOSTMT	NOGRAPHIC	NOIMPRECISE	NOINCLUDE
NOINTERRUPT	NOLIST	NOMAP	NOMARGINI	NONEST
NONUMBER	NOOFFSET	NOOPTIMIZE	NOSTORAGE	NOSYNTAX(S)
NOTERMINAL	NOXREF	OBJECT	ODECK	SEQUENCE(73,80)
SIZE(506756)	SOURCE	STMT		

The installation CLIST tailors the PL/I sample programs for either OS PL/I Version 1, OS PL/I Version 2, or IBM PL/I for MVS & VM compilers, depending on the fields you entered in panel DSNTIPG. To use IMS from PL/I Version 2 Release 3 or IBM PL/I for MVS & VM, you must use the SYSTEM(IMS) compile-time option when compiling your application program. In addition, you must also specify entry point PLISTART when your application is link-edited. For more information, see *OS PL/I Programming Guide* or *IBM PL/I MVS & VM Programming Guide* as appropriate.

Phase 0: Deleting the Sample Objects (DSNTEJ0)

Phase 0 consists of one job, DSNTEJ0. It frees all plans, drops all objects, and deletes data sets so that Phase 1 can be run again. Run Phase 0 (job DSNTEJ0) only if you want to remove all the verification processing you have done so far so you can begin the verification procedure again. When you complete the verification procedure, save the verification objects; you need them when you migrate to the next release of DB2.

If a sample application abends while running a utility, ensure that the utility is terminated before attempting to rerun the job. For information on the -TERM UTILITY command, see Chapter 2 of *Command Reference*.

Even when DSNTEJ0 runs successfully, some of the FREE, DROP, and DELETE commands often fail because the object was not created earlier. You can ignore these errors even though they might generate return codes of 8 or 12. Check other errors.

If DSNTEJ0 runs successfully, it produces the return codes shown in Table 75 on page 2-313.

Table 75. DSNTEJ0 Return Codes

Step	PROCSTEP	Return code
PH00S01		0000, 0008, or 0012
PH00S02		0000 or 0008

If this job fails or abends, be sure that the user specified on the JOB statement is an authorized ID. If the name you specified for either SYSTEM ADMIN 1 or SYSTEM ADMIN 2 on installation panel DSNTIPP is a primary authorization ID, use this name. If the sample authorization exit and RACF are installed, and the SYSTEM ADMIN 1 and SYSTEM ADMIN 2 are known to DB2 as secondary authorization IDs, you can run these jobs under a user ID in either of these RACF groups. Then correct any other problems, and rerun the job from the last successful step.

If the subsystem data sets were deleted before the DB2 sample objects are deleted, you must delete the data sets using access method services or TSO commands. In all of the following examples, *vcatalog* is the catalog alias name you specified for the CATALOG ALIAS field on installation panel DSNTIPA2.

The following access method services commands, which can be executed under TSO, delete the Version 5 sample data sets:

```
DELETE 'vcatalog.DSNDBC.DSN8D51A.*.I0001.A001'
DELETE 'vcatalog.DSNDBC.DSN8D51P.*.I0001.A001'
DELETE 'vcatalog.DSNDBC.DSN8D51A.XEMPL1.I0001.*'
DELETE 'vcatalog.DSNDBC.DSN8D51A.DSN8S51E.I0001.*'
DELETE 'vcatalog.DSNDBC.DSN8D51U.*.I0001.A001'
```

Phase 1: Creating and Loading Sample Tables

This phase consists of two jobs: DSNTEJ1 and DSNTEJ1P. DSNTEJ1 invokes program DSNTIAD, which creates objects during the verification procedure. DSNTEJ1P prepares and invokes program DSNTEP2, which lists contents of the sample tables.

Run DSNTEJ1 before running any other sample jobs. You cannot run DSNTEJ1P if you do not have PL/I, but you can run the queries in job DSNTEJ1P with the SPUFI facility of DB2I.

Job DSNTEJ1

Job DSNTEJ1 consists of the following steps:

Table 76 (Page 1 of 2). Steps in Job DSNTEJ1

Step	Function
1-4	Creates all objects (storage group, databases, table spaces, tables, indexes, and views) used by the samples.
5	Drops synonyms

Table 76 (Page 2 of 2). Steps in Job DSNTEJ1

Step	Function
6	Creates synonyms and grants authorization on objects to PUBLIC AT ALL LOCATIONS. This step creates synonyms for the sample tables, indexes, and views, so that the currently running authorization ID can execute the sample application and grant appropriate authority. The sample dynamic SQL program DSNTIAD is used to process the DB2 object definitions in this step and several others.
7	Uses the ASMCL procedure to create DSN8EAE1, an edit exit routine.
8	Does an ASM LKED for DSNHUFF.
9	Does an ASM LKED for DSN8FPRC, a sample field procedure.
10	Prepares the sample call attachment facility assembler interface. You must link-edit ISPLINK, the ISPF interface module, with this CAF sample load module. To do this, be sure the link-edit SYSLIB statement that gets the ISPF load module library in procedure DSNHASHM is not commented out.
11	Loads the programming-related tables using the LOAD utility.
12	Loads the sample tables using the LOAD utility.
13	Checks data for referential integrity.
14	Establishes a quiesce point using both log and image copies.
15	Makes an image copy of all the sample tables using the copy facility.
16	Establishes another quiesce point using only image copies.
17	Reorganizes a table space and compiles statistics on all table spaces using the REORG and RUNSTATS utilities.
18	Performs a REORG TABLESPACE with SHRLEVEL CHANGE
19	Loads the sample tables using the LOAD utility.
20	Set CURRENT RULES register and add a check constraint using ALTER TABLE.
21	Checks data for referential integrity.
22	Checks data for check integrity.
23-26	Performs the operations in steps 13-16 except for the REORG on partition 3 of the Employee table space.

If DSNTEJ1 runs successfully, it produces the return codes shown in Table 77.

Table 77 (Page 1 of 2). DSNTEJ1 Return Codes

Step	PROCSTEP	Return Code
PH01S01		0000
PH01S02		0000
PH01S03		0000
PH01S04		0000
PH01S05		0000
PH01S06		0000
PH01S07	ASM	0000
	LKED	0000

Table 77 (Page 2 of 2). DSNTEJ1 Return Codes

Step	PROCSTEP	Return Code
PH01S08	ASM	0000
	LKED	0000
PH01S09	ASM	0000
	LKED	0000
PH01S10	PC	0004
	ASM	0000
	LKED	0000
PH01S11	DSNUPROC	0000
PH01S12	DSNUPROC	0004
PH01S13	DSNUPROC	0000
PH01S14	DSNUPROC	0000
PH01S15	DSNUPROC	0000
PH01S16	DSNUPROC	0000
PH01S17	DSNUPROC	0000 or 0004
PH01S18		0000
PH01S19	DSNUPROC	0004
PH01S20		0004
PH01S21	DSNUPROC	0004
PH01S22		0000
PH01S23	DSNUPROC	0000
PH01S24	DSNUPROC	0000
PH01S25	DSNUPROC	0000
PH01S26	DSNUPROC	0000

DB2 issues the following message

DSNT400I SQLCODE = 0, SUCCESSFUL EXECUTION

for every SQL statement, except for the drop synonym and insert statements. If the synonyms in the drop synonym statements are not defined, SQL return codes of -204 result. The insert statements violate a check constraint on the EMP table. This results in a SQL return code of -545.

Job DSNTEJ1P

DSNTEJ1P precompiles, compiles, and link-edits PL/I program DSNTEP2. This program is then used to list the sample database tables and views. It is a dynamic PL/I program that accepts SQL statements. It produces a listing of the results of SELECT statements.

If DSNTEJ1P runs successfully, it produces the return codes shown in Table 78.

Table 78. DSNTEJ1P Return Codes

Step	PROCSTEP	Return code
PH01PS01	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0000
PH01PS02		0000

You can compare the output from this job with the sample output found in “Appendix A. Sample Application Output” on page X-3.

If either DSNTJ1 or DSNTJ1P fails or abends, be sure that the USER specified in the JOB statements is an authorized ID. If the name you specified for either SYSTEM ADMIN 1 or SYSTEM ADMIN 2 on installation panel DSNTIPP is a primary authorization ID, use this name. If the sample authorization exit and RACF are installed, and the SYSTEM ADMIN 1 and SYSTEM ADMIN 2 are known to DB2 as secondary authorization IDs, you can run these jobs under a user ID in either of these RACF groups.

Then, correct any other problems. Before rerunning DSNTJ1, run DSNTJ0 to drop the sample data. If you rerun DSNTJ1P, rerun it from the last successful step.

Phase 2: Testing the Batch Environment

This phase consists of several jobs. Run the jobs to test the program preparation procedures for various languages.

If any of the Phase 2 jobs fail or abend, be sure that the user specified in the JOB statements is authorized. Use the name you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP. Then correct any other problems, and rerun the jobs from the last successful step.

Job DSNTJ2A

DSNTJ2A tests the assembler program preparation procedures. This job prepares and invokes program DSNTIAUL, which demonstrates the use of dynamic SQL in assembler to unload the data from tables or views. It also generates LOAD utility statements so the data can be loaded into another table. DSNTJ2A then uses the LOAD utility to put data into copies of the unloaded tables.

If DSNTJ2A runs successfully, it produces the return codes shown in Table 79. Note that the output from Job DSNTJ2A shown in Table 79 assumes that you are using IBM High Level Assembler/MVS Version 1 Release 2. With Version 2, you can get a return code of 4 at procstep ASM along with warning and informational messages from the Assembler. These warnings relate to changes in the assembler default options and can be ignored. Job DSNTJ2A runs as expected.

#

Table 79. DSNTJ2A Return Codes

Step	PROCSTEP	Return code
PREPUNL	PC	0000
	ASM	0000
	LKED	0000
BINDUNL		0000
DELETE		0000
CREATE		0000
UNLOAD		0000
EDIT		0000
LOAD	DSNUPROC	0004

You can compare the output from this job with the sample output found in “Appendix A. Sample Application Output” on page X-3.

Job DSNTEJ2C

Job DSNTEJ2C tests the COBOL program preparation procedures. This job runs the phone application.

The phone application processes a table of telephone numbers, executing various types of SELECT statements and producing the corresponding listings. It can also update a phone number. This program is discussed in detail in “The Phone Application Scenario” on page 2-351.

COBOL programs can use the VS COBOL II or IBM COBOL compilers, or even OS/VS COBOL. To run DSNTEJ2C with a type of COBOL other than the one you selected on panel DSNTIPY, see “Special Considerations for COBOL Programs” on page 2-308. Be aware that you must use the same type of COBOL to prepare DSNTEJ2C, DSNTEJ3C, DSNTEJ4C, AND DSNTEJ5C.

If DSNTEJ2C runs successfully, it produces the return codes shown in Table 80.

Table 80. DSNTEJ2C Return Codes

Step	PROCSTEP	Return code
PH02CS01		0000
PH02CS02	PC	0004
	COB/COB2/IBMCOB	0000 or 0004
	PLKED ¹	0004
	LKED	0004
PH02CS03	PC	0000
	COB/COB2/IBMCOB	0000 or 0004
	PLKED ¹	0004
	LKED	0000
PH02CS04		0000

¹Only for IBM COBOL

You can compare the output from this job with the sample output found in “Appendix A. Sample Application Output” on page X-3.

Job DSNTEJ2D

Job DSNTEJ2D tests the C program preparation procedures. You must have sequence numbering on to run this job from an ISPF session. The C job runs only the phone application. See the phone application description under DSNTEJ2C on page 2-317.

If DSNTEJ2D runs successfully, it produces the return codes shown in Table 81.

Table 81 (Page 1 of 2). DSNTE2D Return Codes

Step	PROCSTEP	Return code
PH02DS01	PC	0004
	C	0000
	PLKED	0000 or 0004
	LKED	0004

Table 81 (Page 2 of 2). DSNTE2D Return Codes

Step	PROCSTEP	Return code
PH02DS02	PC	0000
	C	0000
	PLKED	0000 or 0004
	LKED	0000 or 0004
PH02DS03		0000

You can compare the output from this job with the sample output found in “Appendix A. Sample Application Output” on page X-3.

Job DSNTEJ2E

Job DSNTEJ2E tests the C++ program preparation procedures. You must have sequence numbering on to run this job from an ISPF session. The C++ job runs only the phone application, “Job DSNTEJ2C” on page 2-317.

If DSNTEJ2E runs successfully, it produces the return codes shown in Table 82.

Table 82. DSNTE2E Return Codes

Step	PROCSTEP	Return code
PH02ES01	PC	0004
	C	0000
	PLKED	0000 or 0004
	LKED	0004
PH02ES02	PC	0000
	CP	0000
	PLKED	0000 or 0004
	LKED	0004
PH02ES03	PC	0004
	CP	0000
	PLKED	0000 or 0004
	LKED	0000
PH02ES04		0000

You can compare the output from this job with the sample output found in “Appendix A. Sample Application Output” on page X-3.

Job DSNTEJ2F

Job DSNTEJ2F tests the FORTRAN program preparation procedures. The FORTRAN job runs only the phone application. See the phone application description under DSNTEJ2C on page 2-317.

If DSNTEJ2F runs successfully, it produces the return codes shown in Table 83.

Table 83 (Page 1 of 2). DSNTE2F Return Codes

Step	PROCSTEP	Return code
PH02FS01	PC	0004
	ASM	0000
	LKED	0004
PH02FS02	PC	0000
	FORT	0000
	LKED	0000
PH02FS03		0000

You can compare the output from this job with the sample output found in “Appendix A. Sample Application Output” on page X-3.

Job DSNTEJ2P

Job DSNTEJ2P tests the PL/I program preparation procedures. The PL/I job runs the phone application.

If DSNTEJ2P runs successfully, it produces the return codes shown in Table 84.

Table 84. DSNTEJ2P Return Codes

Step	PROCSTEP	Return code
PH02PS01		0000
PH02PS02	PPLI	0000
	PC	0004
	PLI	0004
	LKED	0004
PH02PS03	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0000
PH02PS04		0000
PH02PS05		0000

You can compare the output from this job with the sample output provided in “Appendix A. Sample Application Output” on page X-3.

Phase 3: Testing SPUFI, DRDA Access, Dynamic SQL, and TSO

Phase 3 allows you to test SPUFI and DRDA access, run dynamic SQL statements, run the phone application in TSO, and bind packages at the local and remote locations.

SPUFI (SQL Processor Using File Input) is a facility of DB2I. “Testing SPUFI” provides instructions for testing it. You can only run SPUFI under ISPF. You can run dynamic SQL whether or not you have ISPF.

Testing SPUFI

You can test SPUFI by following the steps below:

1. Log on to TSO.
2. Enter ISPF (this might be done for you, depending on your site's standard practice).
3. Select DB2I on the ISPF Primary Option Menu.
4. Select SPUFI on the DB2I menu.
5. Enter the library name *prefix*.NEW.SDSNSAMP(DSNTEJ2P) as input to SPUFI on line 1, the DATASET NAME parameter. If your site uses the comma as a decimal point, the library name entered must be for the tailored version of job DSNTEJ2P that was modified by the installation CLIST.
6. Define an output data set name on line 4, the output DATASET NAME parameter of the panel. This allows you to review the output.

7. Press ENTER, and examine the results. These SQL statements require a significant amount of DB2 processing; you could have to wait for the output.

Run steps 5, 6, and 7 three times:

- Once with member DSNTESA, which uses a set of SQL statements to create a temporary table space and table (as discussed in “Dynamic SQL Statements (DSNTESA, DSNTESQ)” on page 2-363)
- Once with member DSNTESC, which drops table spaces, creates a table space, and creates a PLAN_TABLE in that table space (as discussed in Section 5 (Volume 2) of *Administration Guide*).
- Once with member DSNTese, which retrieves the EXPLAIN information.

If any step fails or abends, be sure that the DB2 subsystem name is specified in the DB2 NAME field on the DB2I Defaults panel.

If you are installing or migrating from Version 4, DSNTESC might fail with a return code of 0008 on either or both of the SQL statements that drop DSN8D51A.DSN8S51X and DSN8D41A.DSN8D41X if these objects did not exist previously. If this occurs, you must change the failing statements to comments. To do this, insert double dashes (--) before each line of the statement.

Also, make sure that the USERID you are using is authorized. If the name you specified for either SYSTEM ADMIN 1 or SYSTEM ADMIN 2 on installation panel DSNTIPP is a primary authorization ID, use this name. If the sample authorization exit and RACF are installed, and both SYSTEM ADMIN 1 and SYSTEM ADMIN 2 are known to DB2 as secondary authorization IDs, you can run these jobs under a user ID in either of these RACF groups. Then correct any other problems and rerun the scenario from the last successful step.

Running SPUFI at Remote non-DB2 Systems

You can use SPUFI to execute an interactive CONNECT statement and then execute SQL statements at a remote location. To do that on non-DB2 systems, you must bind a package for SPUFI on each of those systems. Use the following commands:

```
BIND PACKAGE (location_name.DSNESPCS) MEMBER(DSNESM68)
           ACTION(ADD) ISOLATION(CS) LIB('prefix.SDSNDBRM')
```

```
BIND PACKAGE (location_name.DSNESPRR) MEMBER(DSNESM68)
           ACTION(ADD) ISOLATION(RR) LIB('prefix.SDSNDBRM')
```

If the BIND PACKAGE command fails, the package already exists. See if the time and date formats returned by the existing packages are satisfactory. If they are, the existing packages can be used without any change to the package list in the SPUFI plans. If you need to change the time and date formats returned by the existing packages, you must bind new packages with different collection identifiers that have been agreed to by the application server.

For example, if the collection identifiers are PRIVATCS and PRIVATRR, the commands for doing a remote bind are as follows:


```
BIND PACKAGE (location_name.PRIVATCS) MEMBER(DSNESM68)
ACTION(ADD) ISOLATION(CS) LIB('prefix.SDSNDBRM')
```

```
BIND PACKAGE (location_name.PRIVATRR) MEMBER(DSNESM68)
ACTION(ADD) ISOLATION(RR) LIB('prefix.SDSNDBRM')
```

The SPUFI plans at the DB2 system must be rebound because the location name parameter (which is usually optional) must be explicitly specified for the remote access functions to construct the correct package name. (SPUFI does not use the SQL statement SET CURRENT PACKAGESET.) The location name entry in the package list must precede any pattern-matching character entry. For example, the package list for the DSNESPCS plan is as follows:

```
location_name.PRIVATCS.DSNESM68
*.DSNESPES.DSNESM68
```

The package list for the DSNESPRR plan is as follows:

```
location_name.PRIVATRR.DSNESM68
*.DSNESPES.DSNESM68
```

Running Dynamic SQL and the ISPF/CAF Application

The Phase 3 jobs install the ISPF/CAF sample application. This sample consists of an assembler or COBOL call attachment facility (CAF) interface, a connection manager program, the phone application, and the distributed application using DRDA access. Job DSNTEJ1 prepares the assembler interface, and job DSNTEJ3C prepares the COBOL interface. The connection manager program and the phone application each exist in COBOL and PL/I. Job DSNTEJ3C prepares the COBOL version; job DSNTEJ3P prepares the PL/I version. The distributed application using DRDA access is written in COBOL.

COBOL programs can use the VS COBOL II or IBM COBOL for MVS & VM compilers, or even OS/VS COBOL. To run DSNTEJ3C with a type of COBOL other than the one you specified on field 2 of panel DSNTIPY, see “Special Considerations for COBOL Programs” on page 2-308. Remember that you must use the same type of COBOL to prepare DSNTEJ2C, DSNTEJ3C, DSNTEJ4C, and DSNTEJ5C.

Jobs DSNTEJ3C and DSNTEJ3P:

To prepare for the distributed sample application, DSNTEJ3C binds a package at the local and remote subsystems. The remote subsystem is at the location specified on installation panel DSNTIPY. This allows you to access data at either site. Both the local and remote systems must be running DB2 Version 5.

Because DSNTEJ3C does a remote bind, you must set up your local and remote systems for remote communication before running this job. The sample jobs DSNTEJ1 and DSNTEJ6 must have been run on the remote system. For concurrent installations at 2 DB2 locations, designate one location as the requester and the other location as the server. For information on how to set up DB2 for remote communication, see “Chapter 3-1. Connecting Distributed Database Systems” on page 3-3.

In addition to preparing the phone application, DSNTEJ3P (but not DSNTEJ3C) executes SQL statements dynamically. It executes the same three sets of SQL

statements (DSNTESA, DSNTESC, and DSNTese) explained in “Testing SPUFI” on page 2-319.

If DSNTJ3C runs successfully, it produces the return codes shown in Table 85.

Table 85. DSNTJ3C Return Codes

Step	PROCSTEP	Return code
PH03CS01	PC	0004
	COB/COB2/IBMCOB	0000/0000 or 0004
	PLKED ¹	0004
	LKED	0000
PH03CS02	PC	0004
	COB/COB2/IBMCOB	0004/0000 or 0004
	PLKED ¹	0004
	LKED	0000
PH03CS03	PC	0000
	COB/COB2/IBMCOB	0004/0004
	PLKED ¹	0004
	LKED	0000
PH03CS04	PC	0000 or 0004
	COB/COB2/IBMCOB	0000/0000 or 0004
	PLKED ¹	0004
	LKED	0000
PH03CS05		0000
PH03CS06		0004 or 0000

¹Only for IBM COBOL

Step PH03CS06 can give a return code of 0004 if sample job DSNTJ1 was not run on the remote system. For testing, you should run job DSNTJ1 on the remote system. If DSNTJ3P runs successfully, it produces the return codes shown in Table 86.

Table 86. DSNTJ3P Return Codes

Step	PROCSTEP	Return code
PH03PS01	PPLI	0000
	PC	0004
	PLI	0000
	LKED	0000
PH03PS02	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0000
PH03PS03		0000
PH03PS04		0008 or 0000

The first three steps in job DSNTJ3P, steps PH03PS01, PH03PS02, and PH03PS03, prepare the ISPF/CAF sample application. Step PH03PS04 executes SQL statements dynamically. This step is independent of the first three steps in this job; it might be executed whether or not the first three steps execute successfully.

If you are installing or migrating from DB2 for MVS/ESA Version 4, DSNTESC might fail with a return code of 0008 on either or both of the SQL statements that drop DSN8D51A.DSN8S51X and DSN8D41A.DSN8D41X if these objects did not exist previously. If you get a 0008 return code for any other reason, correct the problem and execute the statements in DSNTESC again.

Starting an Application in an ISPF/TSO Environment

You must have access to ISPF load module libraries to run the ISPF/CAF sample application. See “Making Panels, Messages, and Load Modules Available to ISPF and TSO” on page 2-245 for more information on this procedure. To start the application, enter a CALL command for option 6 of the ISPF primary option menu. To start the COBOL phone sample version of the connection manager, enter:

```
CALL 'prefix.RUNLIB.LOAD(DSN8SCM)'
```

To start the PL/I phone sample version of the connection manager, enter:

```
CALL 'prefix.RUNLIB.LOAD(DSN8SPM)'
```

After you enter one of these commands, DB2 displays the connection manager panel, shown in Figure 63.

```
DB2 SAMPLE APPLICATIONS MENU
====>

Select one of the following options and press enter.

 1 COBOL PHONE SAMPLE      (DB2 ISPF COBOL Application)
 2 PL/I  PHONE SAMPLE      (DB2 ISPF PL/I  Application)
 3 COBOL ORGANIZATION      (DB2 ISPF COBOL Application)

SPECIFY DB2 SUBSYSTEM NAME ====> DSN

PRESS:  END TO EXIT
```

Figure 63. Initial Panel of the ISPF/CAF Application

Choosing option 1 or 2 on the connection manager panel invokes either the COBOL or the PL/I version of the phone application. For more information about the phone application, see “The Phone Application Scenario” on page 2-351. Choosing option 3 on the connection manager panel invokes the COBOL organization application, which uses DRDA access to distributed data. For more information, see “The Distributed Organization Application Scenario” on page 2-355.

Phase 4: Testing the IMS Environment

Phase 4 installs the sample IMS transactions for both COBOL and PL/I. In the PL/I version, the phone application discussed in Phase 2 is also installed as an online transaction. For more information on the phone application, refer to “The Phone Application Scenario” on page 2-351.

Jobs DSNTEJ4C and DSNTEJ4P

Job DSNTEJ4C is for COBOL; DSNTEJ4P is for PL/I. Both jobs perform the following functions:

- Precompile, compile, and link-edit the IMS online applications.
- Bind the IMS online applications.
- Create the message format service (MFS) panels for the online applications.
- Run the required PSBGEN and ACBGEN.

Select the proper job and define the applications and transactions to IMS. Member DSN8FIMS in *prefix*.SDSNSAMP contains information to assist in the definition step.

The verification transactions are single mode, single segment, and nonconversational. We recommend SSM error option R, because the program handles any errors. A resource translation table is not required.

Invoke the transaction by using the FORMAT command. The programs accept several lines of input on the first panel and display the results after you press ENTER.

COBOL programs can use the VS COBOL II or IBM COBOL for MVS & VM or even OS/VS COBOL compilers. To use a type of COBOL other than the one you specified on field 2 of panel DSNTIPY, see "Special Considerations for COBOL Programs" on page 2-308. Remember, you must use the same type of COBOL to prepare DSNTJ2C, DSNTJ3C, DSNTJ4C, and DSNTJ5C.

If DSNTJ4C runs successfully, it produces the return codes shown in Table 87.

Table 87. DSNTJ4C Return Codes

Step	PROCSTEP	Return code
PH04CS01	PC	0004
	COB/COB2/IBMCOB	0000
	PLKED ¹	0004
	LKED	0004
PH04CS02	PC	0000
	COB/COB2/IBMCOB	0000
	PLKED ¹	0004
	LKED	0004
PH04CS03	PC	0000
	COB/COB2/IBMCOB	0000
	PLKED ¹	0004
	LKED	0004
PH04CS04		0000
PH04CS05		0000
PH04CS06	S1	0000
	S2	0004
PH04CS07	S1	0000
	S2	0004
PH04CS08	C	0000
	L	0000
PH04CS09	G	0000

¹Only for IBM COBOL

For DSNTJ4C, the warning code expected from the precompiler step PH04CS01 is:

```
DB2 SQL PRECOMPILER    MESSAGES
DSNH0531 W    NO SQL STATEMENTS WERE FOUND
```

If DSNTJ4P runs successfully, it produces the return codes shown in Table 88 on page 2-325.

Table 88 (Page 1 of 2). DSNTEJ4P Return Codes

Step	PROCSTEP	Return Code
PH04PS01	PPLI	0000
	PC	0004
	PLI	0004
	LKED	0004
PH04PS02	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH04PS03	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH04PS04	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH04PS05	PPLI	0000
	PC	0004
	PLI	0004
	LKED	0004
PH04PS06	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH04PS07	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH04PS08		0000
PH04PS09		0000
PH04PS10	S1	0000
	S2	0004
PH04PS11	S1	0000
	S2	0004
PH04PS12	S1	0000
	S2	0000 or 0004
PH04PS13	S1	0000
	S2	0000 or 0004
PH04PS14	S1	0000
	S2	0000 or 0004
PH04PS15	S1	0000
	S2	0000 or 0004
PH04PS16	C	0000
	L	0000
PH04PS17	G	0000
PH04PS18	C	0000
	L	0000
PH04PS19	G	0000
PH04PS20	C	0000
	L	0000
PH04PS21	G	0000

For DSNTEJ4P, the warning code expected from the precompiler step PH04PS01 is:

```
DB2 SQL PRECOMPILER      MESSAGES
DSNH0531 W   NO SQL STATEMENTS WERE FOUND
```

If either job DSNTEJ4C or job DSNTEJ4P fails or abends, rerun the jobs from the last successful step.

Starting an Application in an IMS Environment

After logging on to IMS, you can start the organization or project application by entering a FORMAT command. The FORMAT commands are:

- /FORMAT DSN8IPGO, which starts the PL/I organization version
- /FORMAT DSN8ICGO, which starts the COBOL organization version.

When you enter either of these two commands, the panel shown in Figure 64 is displayed.

```
MAJOR SYSTEM ...: 0           ORGANIZATION
ACTION .....:
OBJECT .....:
SEARCH CRITERIA.:
DATA .....:
```

Figure 64. Organization Version of Format Command Display

When the following command is entered, the panel shown in Figure 65 is displayed.

- /FORMAT DSN8IPFO, which starts the PL/I projects version.

```
MAJOR SYSTEM ...: P           PROJECTS
ACTION .....:
OBJECT .....:
SEARCH CRITERIA.:
DATA .....:
```

Figure 65. Project Version of Format Command Display

Using the Phone Application in IMS

When you use IMS, information is interactively processed. To begin, clear the screen and type in a FORMAT command. The FORMAT command is:

- /FORMAT DSN8IPNO, which starts PL/I phone application.

When the FORMAT command is entered, the panel shown in Figure 66 on page 2-327 is displayed.

```

----- TELEPHONE DIRECTORY -----

                LAST  NAME ==>

                FIRST NAME ==>

LAST NAME          : *  FOR LIST OF ENTIRE DIRECTORY
                   %  FOR GENERIC LIST (EX. K% = ALL  K - NAMES)
FIRST NAME(OPTIONAL): %  FOR GENERIC LIST

```

Figure 66. Starting the Phone Application

Phase 5: Testing the CICS Environment

Phase 5 tests the CICS environment. It installs the sample applications for COBOL and PL/I, and it prepares the CICS SQLCA formatter front-end.

Job DSNTTEJ5A

DSNTEJ5A assembles and link-edits DSNTIAC, the CICS SQLCA formatter front-end. It also assembles and links the RCT and optionally adds the sample definitions to the CSD. Use DSNTIAC as an alternative to DSNTIAR when you want CICS services to do storage handling and program loading.

Table 89. DSNTEJ5A Return Codes

Step	PROCSTEP	Return code
PH05AS01		0000
PH05AS02		0000
PH05AS03		0000
PH05AS04		0000
PH05AS05		0000
PH05AS06		0000
PH05AS07		0000

Jobs DSNTTEJ5C and DSNTTEJ5P

Job DSNTTEJ5C installs the sample application transactions in COBOL and prepares the organization application. Job DSNTTEJ5P installs the transactions in PL/I and prepares the organization, project, and phone applications.

COBOL programs can use the VS COBOL II or IBM COBOL for MVS & VM or even OS/VS COBOL compilers. To use a type of COBOL other than the one you specified on field 2 of panel DSNTIPY, see “Special Considerations for COBOL Programs” on page 2-308. Remember, you must use the same type of COBOL to prepare DSNTTEJ2C, DSNTTEJ3C, DSNTTEJ4C, and DSNTTEJ5C.

Both phase 5 jobs perform the following functions:

- Compile and link-edit the CICS online applications
- Bind the CICS online applications

- Create the BMS maps for the online applications.

Select the proper job, and define transactions, programs, and BMS maps to CICS.

prefix.SDSNSAMP members DSN8FPPT, DSN8FPCT, and DSN8FRCT contain the respective PPT, PCT, and RCT entries required for the phase 5 applications. These members help you perform the definition step. Make sure that the subsystem ID (SUBID) in the RCT entry matches your DB2 subsystem ID.

If you are running CICS Version 3 Release 1 or higher, DSN8FPPT and DSN8FPCT are obsolete, because resource definition online (RDO) has replaced table entries for transaction and program definitions. Instead, the installation CLIST tailors job DSNTIJSU to define the entries for you.

If DSNTJ5C runs successfully, it produces the return codes shown in Table 90.

Table 90. DSNTJ5C Return Codes

Step	PROCSTEP	Return code
MAPG	ASSEM	0000
MAPD	ASSEM	0000
DSNH		0000
BIND		0000
MAPGP	ASSEM	0000
MAPGL		0000
MAPDP	ASSEM	0000
MAPDL		0000

If DSNTJ5C fails or abends, rerun the job from the last successful step. To receive more prepare-time detail from DSNTJ5C, change the parameters TERM(LEAVE) and PRINT(LEAVE) to TERM(TERM) and PRINT(TERM). See the discussion of the DSNH CLIST in the Chapter 2 of *Command Reference* for more information.

If DSNTJ5P runs successfully, it produces the return codes shown in Table 91 on page 2-329.

Table 91 (Page 1 of 2). DSNTJ5P Return Codes

Step	PROCSTEP	Return code
PH05PS01	ASSEM	0000
PH05PS02	ASSEM	0000
PH05PS03	ASSEM	0000
PH05PS04	ASSEM	0000
PH05PS05	ASSEM	0000
PH05PS06		0004
PH05PS07	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH05PS08		0004

Table 91 (Page 1 of 2). DSNTEJ5P Return Codes

Step	PROCSTEP	Return code
PH05PS09	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH05PS10		0004
PH05PS11	PPLI	0004
	PC	0000
	PLI	0004
	LKED	0004
PH05PS12		0004
PH05PS13	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH05PS14	ASSEM	0000
PH05PS15	ASSEM	0000
PH05PS16		0004
PH05PS17	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH05PS18		0004
PH05PS19	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH05PS20		0004
PH05PS21	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0004
PH05PS22		0000
PH05PS23		0000
PH05PS24	ASSEM	0000
PH05PH25		0000
PH05PS26	ASSEM	0000
PH05PH27		0000
PH05PS28	ASSEM	0000
PH05PH29		0000
PH05PS30	ASSEM	0000
PH05PH31		0000
PH05PS32	ASSEM	0000
PH05PH33		0000
PH05PS34	ASSEM	0000
PH05PH35		0000
PH05PS36	ASSEM	0000
PH05PH37		0000

If DSNTJE5P fails or abends, rerun the job from the last successful step. You might find it convenient to break up DSNTJE5P and run only the unsuccessful steps.

Starting an Application in a CICS Environment

After logging on to CICS, you can start an organization or project application by entering a CICS transaction code. The CICS transaction codes are:

- D8PP, which starts the PL/I project version
- D8PS, which starts the PL/I organization version
- D8CS, which starts the COBOL organization version.

When these transaction codes are entered, the panels shown in Figure 67 and Figure 68 on page 2-331 are displayed.

```
                ACTION SELECTION
MAJOR SYSTEM ...: 0          ORGANIZATION
ACTION .....:
OBJECT .....:
SEARCH CRITERIA.:
DATA .....:
SELECT AN ACTION FROM FOLLOWING LIST

A   ADD (INSERT)
D   DISPLAY (SHOW)
E   ERASE (REMOVE)
U   UPDATE (CHANGE)
```

Figure 67. Initial Panel for the Organization Application in CICS

```
                ACTION SELECTION
MAJOR SYSTEM ...: P          PROJECTS
ACTION .....:
OBJECT .....:
SEARCH CRITERIA.:
DATA .....:
SELECT AN ACTION FROM FOLLOWING LIST

A   ADD (INSERT)
D   DISPLAY (SHOW)
E   ERASE (REMOVE)
U   UPDATE (CHANGE)
```

Figure 68. Initial Panel for the Project Application in CICS

Refer to “Specifying Values in the Sample Application Panels” on page 2-341 for the criteria you need to enter to run the organization and project applications.

Using the Phone Application in CICS

When you use CICS, information is interactively processed. To begin, clear the screen and type in the transaction code:

D8PT

You can change the transaction codes when you install DB2. Check with your system administrator to find out if they have been changed from those shown.

Using CICS Storage-Handling Facilities

To use the CICS storage-handling facilities when running the CICS sample applications, change your DSNTIAR calls to DSNTIAC calls in DSN8MCxx and DSN8MPxx. Then rerun job DSNTEJ5C or job DSNTEJ5P. The calls should look like this:

```
CALL DSNTIAC(EIB,COMMAREA,SQLCA,MSG,LRECL)
```

You must also define DSNTIAC and DSNTIA1 in the CSD.

Phase 6: Accessing Data at a Remote Site

You can use this phase to verify that the features of DRDA access and DB2 private protocol access are working correctly. During this optional phase, you access data at a remote site using multiple sample applications:

- The DRDA access application (DSNTEJ6 in conjunction with DSNTEJ3C)
- The DB2 private protocol access application (user maintained)
- The stored procedure without result set sample (DSNTEJ6S and DSNTEJ6P)
- The stored procedure with result set sample (DSNTEJ6T and DSNTEJ6D)
- The stored procedure for invoking utilities (DSNTEJ6U)
- The stored procedure for IMS Open Database Access (DSNTEJ61 and DSNTEJ62)

The installation CLIST prepares samples DSNTEJ6, DSNTEJ6S, DSNTEJ6P, DSNTEJ6T, and DSNTEJ6D for you only if you specify **YES** or **AUTO** on the DDF startup option of panel DSNTIPR. If you specify **NO**, the installation CLIST does not prepare these samples. In this case, you should not try to run these phase 6 samples.

The installation CLIST prepares the stored procedures samples for DSNTEJ6, DSNTEJ6S, DSNTEJ6P, DSNTEJ6T, and DSNTEJ6D only if you specify a DB2-established stored procedure JCL PROC name (field 2) on panel DSNTIPX. If you replace the default value with blanks on field 2 of this panel, you cannot start the DB2-established stored procedures address space until you update the subsystem parameter.

The installation CLIST prepares the stored procedures samples for DSNTEJ61 and DSNTEJ62 only if you specify a WLM-established stored procedure JCL PROC name (field 1) on panel DSNTIPX. If you replace the default value with blanks on field 1 of this panel, you cannot start the WLM-established stored procedures address space until you update the subsystem parameter.

The installation CLIST tailors the phase 6 sample jobs according to the information you specify in field REMOTE LOCATION (field 1) of panel DSNTIPY. The guidelines for this field are:

- If the field is blank, the installation CLIST does not prepare any of the phase 6 samples and you should not try to run these samples.
- If the value in the field is the same as the location name for the DB2 subsystem you are installing (field 2 of panel DSNTIPR), then the stored procedures samples are prepared and customized for local use. However, the

DRDA access sample is not prepared. This includes DSNTEJ6 and the DRDA access component of DSNTEJ3C.

- If the value in the field is different from the DB2 location name, the installation CLIST prepares the phase 6 samples assuming that the remote location is the *server*, and the local system is the *client*.

If you are installing and testing two DB2 subsystems concurrently, you must designate one as the server and the other as the client. If you change these designations during your testing, your results will be unpredictable. Verify that your VTAM APPL statement has the parameter SYNCLVL=SYNCPT defined. This allows updates at several locations.

DRDA Access Sample

The distributed application using DRDA access is executed as part of Phase 6. The application is prepared in Phase 3 as part of DSNTEJ3C. Before this application can be run correctly as a DRDA access sample, you must run job DSNTEJ6 at both the local and remote sites to tailor the DEPT sample table for use in a distributed environment.

To set up your samples testing for concurrent installations at two DB2 locations, follow these guidelines:

- Designate one location as the requester (the client) and the other location as the server.
- Run the client version of DSNTEJ6 at the client site only; do **not** run the client version of DSNTEJ6 at the server.
- Edit the server version of DSNTEJ6 at the remote server site; do **not** run the server version of DSNTEJ6 at the client.
- Locate the following text in the server version of DSNTEJ6 within step PH06S01:

```
UPDATE DEPT SET LOCATION = (your remote location name) WHERE DEPTNO = 'F22';  
UPDATE DEPT SET LOCATION = (your location name) WHERE LOCATION = ' ';
```

This text should be replaced with:

```
UPDATE DEPT SET LOCATION = (your location name) WHERE DEPTNO = 'F22';  
UPDATE DEPT SET LOCATION = (your remote location name) WHERE LOCATION = ' ';
```

Job DSNTEJ6

Job DSNTEJ6 consists of the following step:

Step	Function
1	Updates the location column in the department table to the sample location entered on installation panel DSNTIPY

If DSNTEJ6 runs successfully, it produces the return code shown in Table 92 on page 2-333.

Table 92. DSNTEJ6 Return Codes

Step	PROCSTEP	Return code
PH06S01		0000

After job DSNTEJ6 has completed successfully, start the distributed application scenario by following the instructions in “Starting an Application in an ISPF/TSO Environment” on page 2-340.

DB2 Private Protocol Access Sample

To test distributed processing that uses DB2 private protocol access, create a job that performs the following functions:

Step	Function
1	Removes objects VPHONE and VEMPLP, which point to local tables, by dropping VPHONE and VEMPLP views or dropping VPHONE and VEMPLP aliases
2	Sets up sample table access by creating aliases VPHONE and VEMPLP, which point to views DSN8510.VPHONE and DSN8510.VEMPLP at a remote location

Notes:

It is assumed that the views DSN8510.VPHONE and DSN8510.VEMPLP and their underlying tables exist at the remote location. If they do not exist, run job DSNTEJ1 to create them.

Step 1 always has one set of SQL statements that fail; it either drops the views or the aliases, but not both.

If you want to point back to the local sample tables after executing this job, run a job that drops the aliases VPHONE and VEMPLP.

After running this job, re-run Phase 2 jobs DSNTEJ2C through DSNTEJ2P. The Phase 2 phone application jobs now reference data at the remote location through aliases.

You could use the following sample JCL statements to perform these functions:

```
//JOB LIB DD DISP=SHR,
//          DSN=prefix.SDSNLOAD
//*
//*      STEP 1 : SET UP THE SAMPLE TABLE ACCESS
//*
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DSN)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51) PARM('RC0') -
LIB('prefix.RUNLIB.LOAD')
END
//SYSIN DD *
DROP VIEW DSN8510.VPHONE;
DROP VIEW DSN8510.VEMPLP;
COMMIT;
DROP ALIAS VPHONE;
DROP ALIAS VEMPLP;
COMMIT;
//*
//*      STEP 2 : SET UP THE SAMPLE TABLE ACCESS
//*
//STEP2 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
```

```

//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DSN)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51) -
LIB('prefix.RUNLIB.LOAD')
END
//SYSIN DD *
CREATE ALIAS VPHONE FOR SAMPLOC.DSN8510.VPHONE;
CREATE ALIAS VEMPLP FOR SAMPLOC.DSN8510.VEMPLP;
/*

```

Stored Procedure Samples

The stored procedure sample applications demonstrate different ways that a stored procedure can be used by a client to issue DB2 commands to a DB2 server. There are several applications discussed; one sample without a result set, one sample with a result set, and one sample for IMS Open Database Access (ODBA) support. The applications prepare and run two programs each:

- A stored procedure that uses the Instrumentation Facility Interface to issue DB2 commands
- A client program that receives DB2 command text, calls the stored procedure to issue the commands, receives the responses from the stored procedure (either in a parameter that is passed back, or in a result set), and prints the results.

Jobs DSNTJ6S, DSNTJ6P, DSNTJ6D, and DSNTJ6T are not edited unless you specify a distributed environment in field 1 of panel DSNTIPR. To run these applications, you must first start the DB2-established stored procedures address space (SPAS). See “Stored Procedures Parameters Panel: DSNTIPX” on page 2-207 for information on generating the JCL to start the SPAS. Before starting the SPAS, you must have Language Environment and either PL/I for MVS or C installed (depending on the job) to run jobs DSNTJ6S, DSNTJ6P, DSNTJ6T, or DSNTJ6D.

Stored Procedure Sample Without Result Set

This application consists of two jobs; DSNTJ6S and DSNTJ6P. Job DSNTJ6S must be run before job DSNTJ6P. To run these jobs, you must have PL/I for MVS installed on your client and server systems in addition to Language Environment.

For concurrent installations at two DB2 locations:

- Run the server version of DSNTJ6S on the server system only; do **not** run the client version of DSNTJ6S on the server
- Run the client version of DSNTJ6P on the client system only; do **not** run the server version of DSNTJ6P on the client

Job DSNTJ6S

Job DSNTJ6S compiles and link-edits the sample stored procedure DSN8EP2. It also updates the SYSIBM.SYSPROCEDURES catalog table with information about the stored procedure. If you have SQL statements in your stored procedure, you must remove the comment character in the JCL from the step that binds the stored procedure package.

You must run job DSNTJ6S at the DB2 server location.

If DSNTEJ6S runs successfully, it produces the return codes shown in Table 93 on page 2-335.

Table 93. DSNTEJ6S Return Codes

Step	PROCSTEP	Return code
PH06SS01		0000 or 0004
PH06SS02	PPLI	0000
	PC	0004
	PLI	0004
	LKED	0000

Job DSNTEJ6P

Job DSNTEJ6P compiles, link-edits, binds, and runs a sample program, DSN8EP1, that invokes the sample stored procedure.

Before you run DSNTEJ6P, run DSNTEJ6S to create the sample stored procedure.

If DSNTEJ6P runs successfully, it produces the return codes shown in Table 94.

Table 94. DSNTEJ6P Return Codes

Step	PROCSTEP	Return code
PH06PS01	PPLI	0000
	PC	0000
	PLI	0004
	LKED	0000
PH06PS02		0004
PH06PS03		0000

Output from a successful execution of DSNTEJ6P lists each DB2 command executed, followed by the messages generated by the DB2 command processor. Sample output from execution of two commands, -DSN1 DISPLAY LOCATION(*) DETAIL and -DSN1 DISPLAY THREAD(*) LOCATION(*) DETAIL, is shown in Figure 69 on page 2-336.

```

*** INPUT STATEMENT:  -DISPLAY LOCATION(*) DETAIL

*** IFI RETURN AREA:
DSNL200I @ DISPLAY LOCATION REPORT FOLLOWS-
LOCATION          PRDID    LINKNAME  REQUESTERS  SERVERS  CONVERSATIONS
SANTA_TERESA_LAB DSN04010 LUND0      0           1         1
DISPLAY LOCATION REPORT COMPLETE

*** INPUT STATEMENT:  -DISPLAY THREAD(*) LOCATION (*) DETAIL

*** IFI RETURN AREA:
DSNV401I @ DISPLAY THREAD REPORT FOLLOWS -
DSNV402I @ ACTIVE THREADS -
NAME      ST A   REQ ID          AUTHID   PLAN     ASID  TOKEN
BATCH    SP *   15 TEJ6P          SYSADM  DSN8EP  0066  61
V429 CALLING STORED PROCEDURE DSN8EP2, LOAD MODULE DSN8EP2
V445-DB2NET.LUND0.AA9CD93122BE=61 ACCESSING DATA FOR SANTA_TERESA_LAB
V447--LOCATION          SESSID          A ST TIME
V448--SANTA_TERESA_LAB 00D7219373DB6551 S4 9504116093056
DISPLAY ACTIVE REPORT COMPLETE
DSN9022I @ DSNVDT '-DSN1 DISPLAY THREAD' NORMAL COMPLETION

```

Figure 69. Sample Output from Job DSNTJ6P

Stored Procedure Sample With Result Set

This application consists of two jobs; DSNTJ6T and DSNTJ6D. Job DSNTJ6T must be run before job DSNTJ6D. You must have C for MVS installed, in addition to Language Environment to run DSNTJ6D and DSNTJ6T.

For concurrent installations at two DB2 locations:

- Run the server version of job DSNTJ6T on the server side only; do **not** run it on the client side
- Run the client version of job DSNTJ6D on the client side only; do **not** run it on the server side

Job DSNTJ6T

Job DSNTJ6T registers, prepares, and binds the sample stored procedure, DSN8ED2, on the server. It also creates a temporary table to receive the IFI output that is returned as a result set.

If DSNTJ6T runs successfully, it produces the return codes shown in Table 95 on page 2-337.

Table 95. DSNTJ6T Return Codes

Step	PROCSTEP	Return code
PH06TS01		0000 or 0004
PH06TS02	PC	0000 or 0004
	C	0000
	PLKED	0000 or 0004
	LKED	0000 or 0004
PH06TS03		0000

Job DSNTEJ6D

Job DSNTEJ6D compiles, link-edits, binds, and runs sample program DSN8ED1, that invokes the sample for using stored procedure result sets.

Before you run DSNTEJ6D, run DSNTEJ6T to create the sample stored procedure.

If DSNTEJ6D runs successfully, it produces the return codes shown in Table 96.

Table 96. DSNTEJ6D Return Codes

Step	PROCSTEP	Return code
PH06DS01	PC	0000
	C	0004
	PLKED	0000 or 0004
	LKED	0000
PH06DS02		0004
PH06DS03		0000

Output from a successful execution of DSNTEJ6D lists each DB2 command executed, followed by the messages generated by the DB2 command processor. Sample output from execution of two commands, -DSN1 DISPLAY ARCHIVE and -DSN1 DISPLAY THREAD(*) DETAIL is shown in Figure 70 on page 2-338.

```

*** Input Statement:
-DISPLAY ARCHIVE
*** A result set was returned by stored procedure DSN8ED2.

```

```

*** IFI return area:
DSNJ322I < DISPLAY ARCHIVE REPORT FOLLOWS-
          COUNT                TIME
          (TAPE UNITS)          (MIN,SEC)
DSNZPARM          3            0,00
CURRENT           3            0,00
=====
ADDR STATUS CORR-ID VOLSER DATASET_NAME
NO TAPE ARCHIVE READING ACTIVITY.
END OF DISPLAY ARCHIVE REPORT.
DSN9022I < DSNJC001 '-DISPLAY ARCHIVE' NORMAL COMPLETION

```

```

*** Input Statement:
-DISPLAY THREAD(*) DETAIL
*** A result set was returned by stored procedure DSN8ED2.

```

```

*** IFI return area:
DSNV401I < DISPLAY THREAD REPORT FOLLOWS -
DSNV402I < ACTIVE THREADS -
NAME      ST A  REQ ID          AUTHID  PLAN    ASID TOKEN
BATCH    SP *   33 DSNTEJ6D    SYSADM  DSN8ED  0028  137
  V429 CALLING PROCEDURE=DSN8ED2          , LOAD MODULE=DSN8ED2 ,
        PROC=V51BSPAS, ASID=002A, WLM_ENV=
  V445-DB2NET.LUND0.ADAA9AD1C468=137 ACCESSING DATA FOR
        SANTA_TERESA_LAB
  V447--LOCATION          SESSID          A ST TIME
  V448--SANTA_TERESA_LAB 00D72193D5EB24A3  S4 9629811542567
DISPLAY ACTIVE REPORT COMPLETE
DSN9022I < DSNVDT '-DISPLAY THREAD' NORMAL COMPLETION

```

```

*** DSN8ED1 completed; highest return code was 0

```

Figure 70. Sample Output from Job DSNTEJ6D

Sample utilities stored procedure

```

# The DSNTUTILS stored procedure enables execution of DB2 utilities from a DB2
# application program using the SQL CALL statement. When called, DSNUTILS
# dynamically allocates the specified datasets, creates the utility input stream
# (SYSIN), invokes DB2 utilities (DSNUTILB), deletes all rows currently in the global
# temporary table (SYSIBM.SYSPRINT), captures the utility output stream
# (SYSPRINT), and puts this output into the global temporary table
# (SYSIBM.SYSPRINT).

```

```

# The DSNUTILS stored procedure must run as a WLM-managed stored procedure.

```

Job DSNTJ6U

Job DSNTJ6U compiles, link-edits, binds, and runs sample PL/I program
DSN8EPU, which invokes the DSNUTILS stored procedure to execute a utility.

Before you run DSNTJ6U, verify that the DSNUTILS stored procedure was
successfully created in job DSNTIJSJG.

Table 97. DSNTJ6U Return Codes

# Step	PROCSTEPT	Return code
# PH06US01	PC	0000
#	PLI	0004
#	LKED	0000
# PH06US02		0004
# PH06US03		0000

Output from a successful execution of DSNTJ6U lists the parameters specified
followed by the messages generated by the DB2 DIAGNOSE DISPLAY MEPL
utility. For more details on utilities and invoking utilities as a stored procedure, see
Appendix B of *Utility Guide and Reference*.

Sample IMS ODBA stored procedure

IMS Open Database Access (ODBA) support allows a DB2 stored procedure to
directly connect to an IMS DBCTL system and issue DL/I calls to access IMS
databases. A stored procedure can issue database DL/I requests via a new callable
interface. IMS ODBA requires IMS Version 6 or above. For more information on
IMS ODBA, see *Application Programming and SQL Guide*.

This application consists of two jobs; DSNTJ61 and DSNTJ62. Job DSNTJ61
must be run before DSNTJ62. You must have COBOL for MVS and VM and
Language environment installed to run DSNTJ61 and DSNTJ62. You must start
a WLM-established stored procedure address space to run DSNTJ61 and
DSNTJ62. You need to update the startup procedure for the WLM-established
stored procedure address space to add the ODBA data set names to the STEPLIB
and DFSRESLB concatenations. An example of a data set name for ODBA is:

high.level.qualifier.CRESLIB

For more information on the setting up a WLM-established stored procedure
address space see *Application Programming and SQL Guide*.

Job DSNTJ61

Job DSNTJ61 prepares a sample stored procedure DSN8EC1, that uses ODBA.
DSN8EC1 can add, update, delete, and display telephone directory records from
the IMS sample database, DFSIVD1. DSN8EC1 shows how the AERTDLI API is
used to issue IMS DL/I calls.

If DSNTJ61 runs successfully, it produces the return codes shown in Table 98 on
page 2-340.

Table 98. DSNTEJ61 Return Codes

Step	PROCSTEP	Return code
PH061S01		0000 or 0004
PH061S02	PC	0004
	COB	0000
	PLKED	0004
	LKED	0000

Job DSNTEJ62

Job DSNTEJ62 prepares and invokes the sample client program DSN8EC2, which
calls stored procedure DSN8EC1. DSN8EC2 calls the stored procedure DSN8EC1
multiple times to add, delete, and display telephone directory records.

If DSNTEJ62 runs successfully, it produces the return codes shown in Table 99.

Table 99. DSNTEJ62 Return Codes

Step	PROCSTEP	Return code
PH062S01	PC	0000
	COB	0000
	PLKED	0004
	LKED	0000
PH062S02		0000 or 0004
PH062S03		0000

Starting an Application in an ISPF/TSO Environment

You must have access to ISPF load module libraries in order to run the ISPF/CAF sample application. See “Making Panels, Messages, and Load Modules Available to ISPF and TSO” on page 2-245 for more information on this procedure. To start the application, enter a CALL command from option 6 of the ISPF primary option menu.

To start the COBOL sample version of the connection manager, enter:

```
CALL 'prefix.RUNLIB.LOAD(DSN8SCM)'
```

After you enter this command, DB2 displays the connection manager panel, shown in Figure 71.

```
DB2 SAMPLE APPLICATIONS MENU
====>

Select one of the following options and press enter.

 1 COBOL PHONE SAMPLE      (DB2 ISPF COBOL Application)
 2 PL/I  PHONE SAMPLE      (DB2 ISPF PL/I  Application)
 3 COBOL ORGANIZATION      (DB2 ISPF COBOL Application)

SPECIFY DB2 SUBSYSTEM NAME ====> DSN

PRESS:  END TO EXIT
```

Figure 71. Initial Panel of the ISPF/CAF Application

Choosing option 3 on the connection manager panel invokes the COBOL distributed sample application. For more information about the distributed sample application, see “The Distributed Organization Application Scenario” on page 2-355.

The Sample Applications

This section describes the sample applications. The names of the sample applications have changed for Version 5. Check to make sure you have the authority to run the Version 5 sample programs. For information on granting and revoking DB2 privileges, see Section 3 (Volume 1) of *Administration Guide*.

Brief scenarios describe how to display, update, add, and delete information using the sample applications. Another scenario describes how to view or change information using a combination of organization and project applications. This scenario contains problem-solving exercises based upon creating and staffing a new department with new projects.

The output from the install verification steps discussed here appears in “Appendix A. Sample Application Output” on page X-3.

Printing the Sample Application Listings

Most of the DB2 sample applications are contained in *prefix.SDSNSAMP*. The source statements contained in *prefix.SDSNSAMP* can be printed using ISPF facilities, IEBPTPCH, or local facilities. The modules making up the SQLCA formatter routine (DSNTIAR, DSNTIAC, DSNTIA1, and DSNTIAM) are not in the *prefix.SDSNSAMP* library. They are provided in object form in *prefix.SDSNLOAD*.

You might not want to print all members of *prefix.SDSNSAMP* because some of the members are large and contain unprintable data. An alternative is to precompile and compile the wanted program by specifying a cross-reference to the precompiler and compiler. This provides a cross-reference for program variables and is current.

Specifying Values in the Sample Application Panels

You are prompted for the following information when you run the interactive sample applications:

Sample Applications	Distributed Sample Applications
<ul style="list-style-type: none">• MAJOR SYSTEM• ACTION• OBJECT• SEARCH CRITERIA• DATA	<ul style="list-style-type: none">• ACTION• OBJECT• SEARCH CRITERIA• LOCATION• DATA

These categories must be regarded as a family of values that, used together, specify the task to be performed. For MAJOR SYSTEM, ACTION, OBJECT, and SEARCH CRITERIA, a character code of one or two characters is used as a form of shorthand to indicate the desired criteria. The system provides a list of these codes with their meanings. A valid location name of 1 to 16 characters is used for location. The value for data must be consistent with the data type and length of

search criteria. For information on valid location names, see “Chapter 3-1. Connecting Distributed Database Systems” on page 3-3.

MAJOR SYSTEM specifies the major application area. In the sample application, there are two major systems: organization and project. These major systems are implemented in separate transactions to keep the plan sizes reasonable. If you are running the DB2 distributed sample program, organization is the only system; therefore, this criterion is not used.

ACTION specifies what you want to do with the object (specified on another line of the panel). You can display, update, add (insert), or erase (delete) information about the specified object.

OBJECT specifies the object about which you want information. Normally, the action is associated with the object. Examples of objects are information about an employee (EM) or information about the relationship among departments (DS).

Objects can be specified with the following codes for the organization application:

DE Department—general department and manager information for department specified
DS Department structure—hierarchy information for department specified
EM Employee—information concerning employee specified.

Objects can be specified with the following codes for the project application:

PS Project structure—information on projects and subprojects
AL Activity listing—information concerning the different activities that makes up a project
PR Project—general project information
AS Activity staffing—information about the employees staffed for activities of specified projects
AE Activity estimate—information concerning the estimated staffing and time requirements of specified projects.

You are able only to add, update, or erase information about the selected object, although you can search and display based on other criteria. Items that are added or updated can be changed on the screen. Other fields are protected.

SEARCH CRITERIA helps to locate the specific item of information upon which to act. The following codes can be specified for the search criteria field for the organization application:

DI Department number
DN Department name
EI Employee number
EN Employee name
MI Manager number
MN Manager name.

The following codes can be specified for the search criteria for the project application:

DI Department number
DN Department name
EI Employee number
EN Employee name

PI Project number
PN Project name
RI Responsible person number
RN Responsible person name.

LOCATION is used only for the distributed application. It describes the location where the action is to take place. If this criterion is left blank, then the local location is assumed.

DATA further identifies the search criteria target. The data value specified must be consistent with the data type and length of the search criteria code. If the search criterion is an employee name (EN), manager name (MN), or responsible person name (RN), the value of data must be a person's last name. (See Specifying Data Values for additional information on how data values can be specified.)

Data values can be specified using either primary selection or secondary selection. *Primary selection* is the data value itself. Only one set of data values fulfills the request. *Secondary selection* allows multiple sets of data values to fulfill the request. A brief summary of the sets of data values appear on the screen. Each summary has an associated line number. To display additional information about a certain line, enter the line number in the DATA field. Secondary selection allows the application to display a set of values and then provides a prompt to select a specific DATA value. For example, you can display information about a *department (DE)* (the OBJECT) with a *department number (DI)* (the SEARCH CRITERIA) with a DATA value of D11.

Allowable Combinations

The codes cannot be combined indiscriminately. For instance, manager number (MI) is a valid search criterion for a department (DE), but employee number (EI) and project number (PI) cannot be used to locate a department.

You can retrieve data by having the panels prompt you for the proper values. It is not necessary to enter the values one line at a time. If you already know all the values you want, they can be entered at the same time. If the values are only partially entered, you must start with ACTION and enter each value in sequence, not skipping over any values. For example, if you know all the values except OBJECT, only ACTION can be entered. You are prompted for OBJECT. Then you can enter OBJECT, SEARCH CRITERIA, and DATA.

Specifying Data Values

An entry on the DATA field specifies the choice of SEARCH CRITERIA. The values available for DATA are not limited to a select few as are the values for ACTION and OBJECT. There is a wider choice of DATA values and a variety of ways to express them.

If you know only part of a DATA value (for example, you know the department number begins with D), you can specify it as a *pattern*. The pattern can contain any character string with a special meaning, such as:

- The underscore character, `_`, represents any single character.
- The percent character, `%`, represents any string of zero or more characters.

These two special characters can be used in conjunction with other characters to specify a DATA value. Table 100 on page 2-344 demonstrates three ways to use these characters to create a DATA value.

Table 100. Searching for Data Values

Data Value	Search Criteria	Description
%SMITH%	EN (Employee name)	Searches for any last name that contains the word SMITH; for example, BLACKSMITH, SMITHSONIAN, or NESMITHA
E_1	DI (Department number)	Searches for any department number with E in position 1 and 1 in position 3; for example, E71, E21, or EB1
%	Any	All values qualify

The values entered on the SEARCH CRITERIA and DATA fields can choose only one item to be displayed. However, the more usual case is that several items are displayed as a list. When this is the case, a secondary selection can be made by choosing the line number of the item of interest.

Function Keys

The bottom line of the panel displays the function keys that are active for that panel:

Function Key 2—Resend: If the panel is blanked out (for example, you pressed the CLEAR key) or you want to refresh the panel, press function key 2 to return (resend) the display you were viewing to the terminal.

Function Key 3—End: To terminate the application, press function key 3 to clear the screen and continue with other transactions.

Function Key 8—Next: Sometimes a display of information is too large to fit on one panel. Press function key 8 to scroll forward (the lines move upward).

Function Key 10—Left: Press function key 10 to move the field of vision up one level in the department structure. For instance, in Figure 75 on page 2-347, Department E01 is shown on the left, and its subdepartments are shown on the right. When you press function key 10, the screen scrolls so that Department E01 is moved from the left side of the panel to the right side and the department to which it reports appears on the left. All other departments that report to the department now on the left also appear along with Department E01 on the right. Function key 10 performs this function only for IMS and CICS samples.

Scenarios

This section discusses several scenarios for using the sample applications. How you invoke these applications depends on the environment you are working in; instructions appear in the following places:

- “Starting an Application in an ISPF/TSO Environment” on page 2-323 and 2-340
- “Starting an Application in an IMS Environment” on page 2-326
- “Starting an Application in a CICS Environment” on page 2-330.

When an application executes, many areas on the display panel might be highlighted. The data you enter might not be highlighted, depending on the type of panel displayed.

The Project Application Scenario

This scenario demonstrates the use of the project application. For example, you can find the person responsible for a project and list the activities assigned to one of its subprojects. Phase 4 (IMS) and Phase 5 (CICS) prepare the programs that you execute.

After you enter the appropriate transaction code, you see the first panel of the project application. Enter the following values:

- On the MAJOR SYSTEM, enter P for project.
- On the ACTION line, enter D for display.
- On the OBJECT line, enter PS for project structure.
- On the SEARCH CRITERIA line, enter PI for project ID.
- On the DATA line, enter MA2100 as the project ID.

The panel below shows the selected project along with its corresponding subprojects.

```
PROJECT STRUCTURE
MAJOR SYSTEM ...: P          PROJECTS
ACTION .....: D          DISPLAY (SHOW)
OBJECT .....: PS         PROJECT STRUCTURE
SEARCH CRITERIA.: PI      PROJECT ID
DATA .....: MA2100

PROJECT ID & NAME          SUBPROJECT ID & NAME
RESPONSIBLE ID & NAME     RESPONSIBLE ID & NAME

MA2100 WELD LINE AUTOMATION      MA2110 W L PROGRAMMING
000010 CHRISTINE I HAAS         000060 IRVING F STERN

                                PL2100 WELD LINE PLANNING
                                000020 MICHAEL L THOMPSON

PFK: 02=RESEND 03=END 08=NEXT 10=LEFT
```

Figure 72. Project Application—Viewing a Project Structure

Updating an Activity

Suppose you want to update activity information for a project with ID IF1000. Enter the following values:

- On the MAJOR SYSTEM, enter P for project.
- On the ACTION line, enter U for update.
- On the OBJECT line, enter AE for activity estimate.
- On the SEARCH CRITERIA line, enter PI for project ID.
- On the DATA line, enter IF1000 as the project ID.

Press the ENTER key, and a list of project IF1000 activities appears on the panel. Next, choose the activity to be updated. For instance, if you want to update the first activity listed, enter 1 as the DATA value and press the ENTER key. The next

panel shows information about the estimated mean staffing requirements of this activity as well as the start and completion dates. To change information about the estimated end date, enter data over the existing information displayed on that input line. After you have verified the change, press ENTER. The next panel displays the updated information, as shown in Figure 73.

```

                                UPDATING OF AN ACTIVITY ESTIMATE
MAJOR SYSTEM ...: P             PROJECTS
ACTION .....: U             UPDATE (CHANGE)
OBJECT .....: AE            ACTIVITY ESTIMATE
SEARCH CRITERIA : PI         PROJECT ID
DATA .....: 01
DSN8024I DSN8MPX - ACTIVITY SUCCESSFULLY UPDATED
  PROJECT ID           : IF1000
  NAME                 : QUERY SERVICES

  ACTIVITY ID         :          90
      KEYWORD          : ADMQS
      DESCRIPTION      : ADM QUERY SYSTEM
      EST MEAN STAFFING :          2.00
      EST START DATE   : 1982-01-01
      EST END DATE     : 1983-04-15

PFK: 02=RESEND 03=END

```

Figure 73. Project Application—Changes Accepted

To terminate the project application, press the PF3 key. The APPLICATION TERMINATED message is displayed. If you are using CICS, clear the screen and enter a new transaction code. If you are using IMS, clear the screen and enter a new transaction code or a /FORMAT command.

The Organization Application Scenario

This scenario shows how to use the organization application to display a list of departments within a department and the structure of one of these departments. This application is executed in Phase 4 for IMS and Phase 5 for CICS.

After you enter the appropriate transaction code, you see the first panel of the project application. Enter the following values:

- On the MAJOR SYSTEM, enter O for organization.
- On the ACTION line, enter D for display.
- On the OBJECT line, enter DS for department structure.
- On the SEARCH CRITERIA line, enter DI for department number.
- On the DATA line, enter %, which enables you to display a list of all the departments.

Each department entry is numbered on the far left side of the panel as shown in the Figure 74 on page 2-347.

```

DEPARTMENT ADMINISTRATIVE STRUCTURE SELECTION
MAJOR SYSTEM ...: 0      ORGANIZATION
ACTION .....: D      DISPLAY (SHOW)
OBJECT .....: DS      DEPARTMENT STRUCTURE
SEARCH CRITERIA.: DI    DEPARTMENT ID
DATA .....: %
SELECT A DEPARTMENT FROM FOLLOWING LIST BY SPECIFYING THE LINE NUMBER
NO D/ID DEPARTMENT NAME M/ID MANAGER NAME
01 A00 SPIFFY COMPUTER SERVICES DIV. 000010 CI HAAS
02 B01 PLANNING 000020 ML THOMPSON
03 C01 INFORMATION CENTER 000030 SA KWAN
04 D01 DEVELOPMENT CENTER
05 D11 MANUFACTURING SYSTEMS 000060 IF STERN
06 D21 ADMINISTRATION SYSTEMS 000070 ED PULASKI
07 E01 SUPPORT SERVICES 000050 JB GEYER
08 E11 OPERATIONS 000090 EW HENDERSON
09 E21 SOFTWARE SUPPORT 000100 TQ SPENSER
PFK: 02=RESEND 03=END 08=NEXT

```

Figure 74. Organization Application—Viewing a List of Departments

To retrieve further information, specify a line number as a data value. This method is called secondary selection. Secondary selection provides prompts to aid in finding the information to be displayed, added, erased, or updated. If only one entry possibility exists, secondary selection is not offered.

To view an individual department structure, specify the line entry number (secondary selection) of the department as a new DATA value. For example, to view the structure of Department E01, specify a data value of 7 on the DATA entry line (7 is the line number of the entry for Department E01).

The result of entering the data value of 7 is a display of Department E01 and its departments as shown in Figure 75. The department manager for E01 is listed on the left, and the departments of E01 are listed on the right. Employees of E01 are listed below the subdepartments of E01.

```

DEPARTMENT ADMINISTRATIVE STRUCTURE
MAJOR SYSTEM ...: 0      ORGANIZATION
ACTION .....: D      DISPLAY (SHOW)
OBJECT .....: DS      DEPARTMENT STRUCTURE
SEARCH CRITERIA.: DI    DEPARTMENT ID
DATA .....: 07

DEPARTMENT ID & NAME          SUBDEPARTMENT ID, NAME & MANAGER
MANAGER ID & NAME            EMPLOYEE ID & NAME
E01 SUPPORT SERVICES        E11 OPERATIONS
000050 JOHN B GEYER        000090 EILEEN W HENDERSON

                                E21 SOFTWARE SUPPORT
                                000100 THEODORE Q SPENSER

                                000050 JOHN B GEYER

PFK: 02=RESEND 03=END 08=NEXT 10=LEFT

```

Figure 75. Organization Application—Viewing a Department Structure

Starting a New Operation

You can start a new operation on the organization application by moving the cursor to the D on the ACTION line and retaining the D or changing it to a different action (add, erase, or update). Follow the displayed options to perform your selected action.

Alternatively, you can leave the organization application by pressing the PF3 key. If you are using CICS, enter the transaction code. If you are using IMS, clear the screen and enter the /FORMAT command to select the project application. In either case, to proceed with a different operation, select a different ACTION, OBJECT, and so forth.

Adding a New Department

Adding a department falls under the organization major system. Start the organization application as described in “Starting a New Operation” and enter the following values:

- On the MAJOR SYSTEM, enter O for organization
- On the ACTION line, enter A for add (insert)
- On the OBJECT line, enter DE for the department that is to be added
- On the SEARCH CRITERIA line, enter DI for department ID
- On the DATA line, enter C11, the specific department number.

Next, you can enter the details of the new department. The four department fields are department number, department name, manager number, and administration department number. Enter:

- INFORMATION SERVICES for department name
- 000130 for manager number
- C01 for the administration department number.

Press ENTER to display the panel shown in Figure 76. The panel shows the successful addition of the new department.

```
                ADDING A NEW DEPARTMENT
MAJOR SYSTEM ...: O          ORGANIZATION
ACTION .....: A          ADD (INSERT)
OBJECT .....: DE         DEPARTMENT
SEARCH CRITERIA.: DI       DEPARTMENT ID
DATA .....: C11
DSN8012I DSN8MPE - DEPARTMENT SUCCESSFULLY ADDED
DEPARTMENT ID           : C11
NAME                   : INFORMATION SERVICES
MANAGER ID             : 000130
ADMIN DEP ID          : C01
MANAGER ID             : 000130
FIRST NAME            : DOLORES
MIDDLE INITIAL       : M
LAST NAME             : QUINTANA
WORK DEPT ID         : C01

PFK: 02=RESEND 03=END
```

Figure 76. Organization Application—Adding a Department

Deleting an Entry

Deleting an entry in the department table is also a function of the organization major system. Following the process outlined in “Starting a New Operation” on page 2-348, replace the following values on the panel currently displayed on your screen:

- On the MAJOR SYSTEM, enter O for organization.
- On the ACTION line, enter E for erase.
- On the OBJECT line, enter DE for department.
- On the SEARCH CRITERIA line, enter DI for department ID.
- On the DATA line, enter C11 for department name.

Press ENTER to display the panel shown in Figure 77.

```
ERASING A DEPARTMENT
MAJOR SYSTEM ...: O      ORGANIZATION
ACTION .....: E      ERASE (REMOVE)
OBJECT .....: DE     DEPARTMENT
SEARCH CRITERIA.: DI    DEPARTMENT ID
DATA .....: C11
PRESS ENTER TO ERASE A DEPARTMENT
  DEPARTMENT ID      : C11
    NAME             : INFORMATION SERVICES
    MANAGER ID       : 000130
    ADMIN DEP ID     : C01
MANAGER ID          : 000130
  FIRST NAME        : DOLORES
  MIDDLE INITIAL    : M
  LAST NAME         : QUINTANA
  WORK DEPT ID     : C01

PFK: 02=RESEND 03=END
```

Figure 77. Organization Application—Deletion Successful

Press ENTER again to verify the erase action. The following message appears on the panel:

```
DSN8013I    csect DEPARTMENT SUCCESSFULLY ERASED
```

Transferring

The procedure for transferring one employee to another department and replacing that employee involves several steps. In this scenario, John B. Geyer (manager of the department for Support Services) is transferred to the staff of Spiffy Computer Service Division. Bruce Adamson is assigned as manager of Support Services.

To move Adamson into his new position as manager of Support Services, you must determine his employee number. Transferring an employee is a function of the organization major system. Start the organization application as described in “Starting a New Operation” on page 2-348, and enter the following values:

- On the MAJOR SYSTEM, enter O for organization.
- On the ACTION line, enter D for display.
- On the OBJECT line, enter EM for employee.
- On the SEARCH CRITERIA line, enter EN for employee name.
- On the DATA line, enter ADAMSON as the specific employee name.

Press ENTER to display the panel showing that Adamson's employee number is 000150.

The next step is for you to change the manager number for the Support Services department to Adamson's number, 000150. But first you must find the Support Services department. To do this, change ACTION to U (update), OBJECT to DE (department), and SEARCH CRITERIA to DN (department name). Change DATA to %SUPPORT% to specify any department with the word SUPPORT in it.

Press ENTER, and a list of departments with support in their name is displayed. Support Services has line number 01. Enter this number at DATA. (The leading zero is not needed.)

Press ENTER to display the next panel. The only values that can be changed are department name, manager ID, and administration department ID. Enter Adamson's employee number in the Support Services department after MANAGER ID. At this point, the data on the manager still pertains to Geyer.

Press ENTER to display the panel that shows Adamson as manager of Support Services. The work department ID shown (D11) is still Adamson's old number. To change Adamson's work department ID, enter EM for OBJECT, enter EI for SEARCH CRITERIA, and change the employee number to 000150 for DATA.

Press ENTER to display the employee information on Adamson. Now that information on Adamson can be updated. The fields that can be changed are employee first name, middle initial, last name, and work department ID. Enter the middle initial for Adamson, which was not in the database, and the department number E01. Press ENTER, and the information on Adamson is updated, including his new department number.

The final step is to move Geyer to the correct department. Change the SEARCH CRITERIA and DATA to EN and GEYER, respectively. Press ENTER to obtain the next panel. The employee ID, name, and work department ID can be changed on this panel. However, the only change necessary in this case is to change Geyer's work department ID to his new one, A00. The panel in Figure 78 on page 2-351 shows the completed entry.

```

                UPDATING AN EMPLOYEE
MAJOR SYSTEM ...: 0          ORGANIZATION
ACTION .....: U          UPDATE (CHANGE)
OBJECT .....: EM         EMPLOYEE
SEARCH CRITERIA.: EN      EMPLOYEE NAME
DATA .....: GEYER
DSN8004I DSN8MPF - EMPLOYEE SUCCESSFULLY UPDATED
  DEPARTMENT ID      : A00
                    : SPIFFY COMPUTER SERVICE DIV.
  NAME              : 000010
  MANAGER ID        : A00
  ADMIN DEP ID      : 000050
EMPLOYEE ID         : JOHN
  FIRST NAME        : B
  MIDDLE INITIAL    : GEYER
  LAST NAME         : A00
  WORK DEPT ID      : A00

PFK: 02=RESEND 03=END

```

Figure 78. Organization Application—Employee Data Update Completed

To terminate the application and return to the beginning of the operation, press the PF3 key.

The Phone Application Scenario

The phone application is used in phase 2 (batch mode), phase 3 (CAF), and interactively in phase 4 (IMS) and phase 5 (CICS).

The phone application retrieves information from a phone directory and updates employee phone numbers. The phone directory consists of data from a combination (join) of the employee table (DSN8510.EMP) and the department table (DSN8510.DEPT). This joined view is called VPHONE. The program also uses a second view called VEMPLP to update the employee table, which does not affect a view that joins tables.

The phone application is designed to operate in batch and interactively in ISPF/TSO, IMS, and CICS. Table 101 describes the environments in which each phone application operates and the language in which each is written. For information on how to invoke the CAF application in an ISPF/TSO environment, see Figure 63 on page 2-323.

Table 101. Phone Programs

Environment	Language	Name
ISPF/TSO	COBOL	DSN8SC3
ISPF/TSO	PL/I	DSN8SP3
IMS	PL/I	DSN8IP3
CICS	PL/I	DSN8CP3
batch	COBOL	DSN8BC3
batch	FORTRAN	DSN8BF3
batch	PL/I	DSN8BP3
batch	C	DSN8BD3

Phone Application Panels

The panels for the phone application are the same, whether IMS or CICS is used. In both cases, information is managed interactively beginning with the panel shown in Figure 79.

```
----- TELEPHONE DIRECTORY -----  
  
                LAST NAME ==> _  
                FIRST NAME ==>  
  
LAST NAME          * FOR LIST OF ENTIRE DIRECTORY  
                   % FOR GENERIC LIST (EX. K% = ALL K - NAMES)  
FIRST NAME(OPTIONAL) % FOR GENERIC LIST
```

Figure 79. Telephone Application—First Display

On this panel, enter the first and last name of the employee whose telephone number you want to view or change. To see an entire listing of employee numbers, put an * next to the LAST NAME input line. If only part of a first or last name is known, use the percent character (%) to qualify the list of names to appear in the directory. For example, entering K% on the LAST NAME input line calls a list of the telephone numbers of all employees whose last name begins with a K. Similarly, the first name can be qualified.

To keep this sample program as simple as possible and to allow updating, scrolling is not used with the IMS and CICS versions. Scrolling is used with the ISPF/CAF version. Only the first panel of selected names and phone numbers can be displayed. The second panel is the Telephone Directory itself. The employee telephone number is highlighted. To update an employee telephone number, type over the highlighted number and press ENTER. To update a phone number listed under the name Heather A Nicholls, specify NICHOL% when you are not sure if there are one or two Ls in Nicholls.

Press ENTER to display the panel on which the phone number is highlighted. Suppose you want to change the phone number from 1793 to 1795. Just type over the number to be changed. You can type over as many numbers as appear listed in the current display. After you press ENTER, you get a message confirming the updated phone number. The panel in Figure 80 on page 2-353 shows the updated panel.


```

----- TELEPHONE DIRECTORY -----
FIRST NAME MID LAST NAME  PHONE  EMPL WORK WORKDEPT
          INIT          NO    NO  DEPT NAME
HEATHER  A      NICHOLLS   1795 000140 C01 INFORMATION CENTER
.
.
.

```

Figure 80. Telephone Application—Updated Display

Using the Phone Application under Batch

The sample batch phone applications are provided in FORTRAN (DSN8BF3), COBOL (DSN8BC3), C (DSN8BD3), and PL/I (DSN8BP3).

If you want to update an employee phone number, create a data set that contains information about the phone number to be updated. This data set works in combination with another data set that contains JCL for processing information. The first data set consists of card images in the format shown in Table 102.

Table 102. Format of Phone Application Data Set

Column	Description
1	ACTION—U for update, L for list
2	Employee last name
17	Employee first name
29	Employee number
35	New phone number

The ACTION code in this card image indicates whether an employee number is to be updated (U) or listed (L). When updating an employee phone number, only the employee number and the new phone number are specified in the data set. When listing phone numbers, the last name must be specified. Specifying the first name is optional. The * and % can be used with the ACTION code just as they are used with the panels.

Figure 81 on page 2-354 shows an example of an update data set and a list data set. Each time a number is listed or updated, a new data set is created containing a card image like the one in Figure 81. The first card in the data set shows the phone number of employee number 000140 being updated (U) to 6767. The second card shows a list (L) for Heather Nicholls. The last card shows a list (L) of all employees whose first names begin with the letters MAR. The example shows the letters MAR followed by a % in the first name column to indicate that only those employees whose first names begin with MAR are to be listed.

The Distributed Organization Application Scenario

This scenario shows how to use the distributed organization application to display a department structure, display department information, and update a department at a local location. It also shows how to erase and add an employee at a remote location. This application is executed in Phase 6. The application accesses distributed data with DRDA access.

The department information (DEPT table) is shared by all locations. If you make changes to DEPT table at one location, the DEPT tables at the other locations are updated at the same time. The employee information (EMP table) is unique to each location, containing only the employees that work at that particular location.

After you enter the appropriate transaction code, you see the first panel of the organization application.

Displaying Department Structure at the Local Location

To display a department structure, enter the following values:

- On the ACTION line, enter D for display.
- On the OBJECT line, enter DS for department structure.
- On the SEARCH CRITERIA line, enter DI for department number.
- On the LOCATION line, leave blank, indicating local location.
- On the DATA line, enter A00 for department number.

```
DB2 ORGANIZATION APPLICATION

===>_
ACTION .....:d      A (ADD)          E (ERASE)
                   D (DISPLAY)       U (UPDATE)

OBJECT .....:ds     DE (DEPARTMENT)   EM (EMPLOYEE)
                   DS (DEPT STRUCTURE)

SEARCH CRITERIA :di  DI (DEPARTMENT ID)  MN (MANAGER NAME)
                   DN (DEPARTMENT NAME) EI (EMPLOYEE ID)
                   MI (MANAGER ID)    EN (EMPLOYEE NAME)

LOCATION .....:      (Blank implies local location)

DATA .....:a00

PRESS: ENTER to process   END to exit
```

Figure 84. Starting the Distributed Organization Application

Press the ENTER key. The panel below shows the structure of the department requested.

```

                                DB2 ORGANIZATION APPLICATION                ROW 1 of 5
====>
PRESS: ENTER TO PROCESS   END TO EXIT

DEPARTMENT STRUCTURE FOR:

----- DEPARTMENT ID AND NAME----- ----- MANAGER ID AND NAME-----
A00 SPIFFY COMPUTER SERVICE DIV.   000010 CHRISTINE   I HAAS

SUBDEPARTMENTS:

A00 SPIFFY COMPUTER SERVICE DIV.   000010 CHRISTINE   I HAAS
B01 PLANNING                       000020 MICHAEL     L THOMPSON
C01 INFORMATION CENTER             000030 SALLY        A KWAN
D01 DEVELOPMENT CENTER
E01 SUPPORT SERVICES               000050 JOHN        B GEYER

```

Figure 85. Displaying Department Structure

Press ENTER or END to exit.

Displaying Department Information at the Local Location

To display department information, enter the following values:

- On the ACTION line, enter D for display.
- On the OBJECT line, enter DE for department.
- On the SEARCH CRITERIA line, enter DI for department number.
- On the LOCATION line, leave blank, indicating local location.
- On the DATA line, enter A00 for department number.

```

                                DB2 ORGANIZATION APPLICATION
====>_
ACTION .....:d      A (ADD)           E (ERASE)
                   D (DISPLAY)        U (UPDATE)

OBJECT .....:de     DE (DEPARTMENT)    EM (EMPLOYEE)
                   DS (DEPT STRUCTURE)

SEARCH CRITERIA :di  DI (DEPARTMENT ID)  MN (MANAGER NAME)
                   DN (DEPARTMENT NAME) EI (EMPLOYEE ID)
                   MI (MANAGER ID)    EN (EMPLOYEE NAME)

LOCATION .....:      (Blank implies local location)

DATA .....:a00

PRESS: ENTER to process   END to exit

```

Figure 86. Starting the Distributed Organization Application

Press the ENTER key. The panel below shows the department information requested.

```

                                DB2 ORGANIZATION APPLICATION
====> _

                                DISPLAY A DEPARTMENT

DEPARTMENT ID                   : A00
NAME                             : SPIFFY COMPUTER SERVICE DIV.
MANAGER ID                       : 000010
ADMIN DEP ID                     : A00
LOCATION                           :

MANAGER ID                       : 000010
FIRST NAME                       : CHRISTINE
MIDDLE INITIAL                   : I
LAST NAME                        : HAAS
WORK DEPT ID                     : A00

PRESS: ENTER TO PROCESS   END TO EXIT

```

Figure 87. Displaying Department Information

Press ENTER or END to exit.

Updating a Department at the Local Location

To update department information, enter the following values:

- On the ACTION line, enter U for update.
- On the OBJECT line, enter DE for department.
- On the SEARCH CRITERIA line, enter DI for department number.
- On the LOCATION line, leave blank, indicating local location.
- On the DATA line, enter % which enables you to display a list of all the departments.

```

                                DB2 ORGANIZATION APPLICATION
====> _

ACTION .....:u      A (ADD)           E (ERASE)
                   D (DISPLAY)        U (UPDATE)

OBJECT .....:de     DE (DEPARTMENT)    EM (EMPLOYEE)
                   DS (DEPT STRUCTURE)

SEARCH CRITERIA :di  DI (DEPARTMENT ID)  MN (MANAGER NAME)
                   DN (DEPARTMENT NAME)  EI (EMPLOYEE ID)
                   MI (MANAGER ID)       EN (EMPLOYEE NAME)

LOCATION .....:      (Blank implies local location)

DATA .....:%

PRESS: ENTER to process   END to exit

```

Figure 88. Starting the Distributed Organization Application

Press the ENTER key. The panel below lists the departments that can be updated. Select the department to be updated by putting an S in the left margin by the department number.

```

                                DB2 ORGANIZATION APPLICATION          ROW 1 OF 9
====> _
ACTION:  UPDATE A DEPARTMENT

TO SELECT FROM THE LIST PLACE AN S NEXT TO THE DEPARTMENT
PRESS ENTER TO PROCESS OR END TO EXIT

D/ID  DEPARTMENT NAME                M/ID  MANAGER NAME
-----
A00   SPIFFY COMPUTER SERVICE DIV.    000010  CI HAAS
B01   PLANNING                        000020  ML THOMPSON
C01   INFORMATION CENTER              000030  SA KWAN
D01   DEVELOPMENT CENTER
D11   MANUFACTURING SYSTEMS          000060  IF STERN
D21   ADMINISTRATION SYSTEMS         000070  ED PULASKI
S E01  SUPPORT SERVICES                000050  JB GEYER
E11   OPERATIONS                     000090  EW HENDERSON
E21   SOFTWARE SUPPORT                000100  TQ SPENSER

```

Figure 89. Selecting a Department to be Updated

Press the ENTER key. The panel below displays the information relevant to the selected department. Enter the information you want to update on this panel; in this case, enter the name of the department.

```

                                DB2 ORGANIZATION APPLICATION
====> _

                                UPDATE A DEPARTMENT

DEPARTMENT ID                   : E01
NAME                            : hardware support service
MANAGER ID                      : 000050
ADMIN DEP ID                    : A00
LOCATION                          :

MANAGER ID                      : 000050
FIRST NAME                      : JOHN
MIDDLE INITIAL                  : B
LAST NAME                       : GEYER
WORK DEPT ID                   : E01

PRESS: ENTER TO PROCESS      END TO EXIT

```

Figure 90. Updating a Department

Press the ENTER key to process the updated information. A message appears on this panel that states the update was successful.

```

                                DB2 ORGANIZATION APPLICATION
====>
DSN8014I  DSN8HC3-DEPARTMENT SUCCESSFULLY UPDATED
                                UPDATE A DEPARTMENT

DEPARTMENT ID                    : E01
NAME                             : HARDWARE SUPPORT SERVICE
MANAGER ID                       : 000050
ADMIN DEP ID                     : A00
LOCATION                           :

MANAGER ID                       : 000050
FIRST NAME                       : JOHN
MIDDLE INITIAL                   : B
LAST NAME                        : GEYER
WORK DEPT ID                     : E01

PRESS: ENTER TO PROCESS  END TO EXIT

```

Figure 91. Update Successfully Processed

Press ENTER to return to the previous panel or END to exit. If you return to the previous panel, you can now select another department to update, or press ENTER or END to exit. The same message appears on this panel, indicating the update was successful.

```

                                DB2 ORGANIZATION APPLICATION      ROW 1 OF 9
====>
DSN8014I  DSN8HC3-DEPARTMENT SUCCESSFULLY UPDATED
ACTION:  UPDATE A DEPARTMENT

TO SELECT FROM THE LIST PLACE AN S NEXT TO THE DEPARTMENT
PRESS ENTER TO PROCESS OR END TO EXIT

D/ID  DEPARTMENT NAME                M/ID  MANAGER NAME
-----
A00   SPIFFY COMPUTER SERVICE DIV.   000010  CI HAAS
B01   PLANNING                       000020  ML THOMPSON
C01   INFORMATION CENTER             000030  SA KWAN
D01   DEVELOPMENT CENTER
D11   MANUFACTURING SYSTEMS         000060  IF STERN
D21   ADMINISTRATION SYSTEMS        000070  ED PULASKI
E01   HARDWARE SUPPORT SERVICE       000050  JB GEYER
E11   OPERATIONS                     000090  EW HENDERSON
E21   SOFTWARE SUPPORT               000100  TQ SPENSER

```

Figure 92. Department Successfully Updated

Adding an Employee at a Remote Location

To add an employee at a remote location, enter the following values:

- On the ACTION line, enter A for add.
- On the OBJECT line, enter EM for employee.
- On the SEARCH CRITERIA line, enter EI for employee number.
- On the LOCATION line, enter *your server location* for the remote location.
- On the DATA line, enter SJ0100 which indicate the employee ID of the employee to be added.

```

                                DB2 ORGANIZATION APPLICATION
====> _

ACTION .....:a      A (ADD)          E (ERASE)
                   D (DISPLAY)       U (UPDATE)

OBJECT .....:em     DE (DEPARTMENT)   EM (EMPLOYEE)
                   DS (DEPT STRUCTURE)

SEARCH CRITERIA :ei  DI (DEPARTMENT ID) MN (MANAGER NAME)
                   DN (DEPARTMENT NAME) EI (EMPLOYEE ID)
                   MI (MANAGER ID)     EN (EMPLOYEE NAME)

LOCATION .....:your server location (Blank implies local location)

DATA .....:sj0100

PRESS: ENTER to process   END to exit

```

Figure 93. Starting the Distributed Organization Application

Press the ENTER key. The panel below allows you to only enter information about the employee. The department information on the panel is protected. Enter the necessary information about the employee.

```

                                DB2 ORGANIZATION APPLICATION
====> _

                                ADD    AN EMPLOYEE

DEPARTMENT ID                   :
NAME                             :
MANAGER ID                       :
ADMIN DEP ID                     :
LOCATION                           :

EMPLOYEE ID                      : SJ0100
FIRST NAME                       : W
MIDDLE INITIAL                   :
LAST NAME                        : WALTERS
WORK DEPT ID                     : F22

PRESS: ENTER TO PROCESS   END TO EXIT

```

Figure 94. Employee to be Added

Press the ENTER key to process or END to exit. If you press the ENTER key, a message appears on the panel indicating that the employee has been added.


```

                                DB2 ORGANIZATION APPLICATION
====>_
DSN8002I   DSN8HC3-EMPLOYEE SUCCESSFULLY ADDED
                                ADD      AN EMPLOYEE

DEPARTMENT ID                    : F22
NAME                              : SPIFFY COMPUTER SERVICE DIV.
MANAGER ID                        :
ADMIN DEP ID                      : E01
LOCATION                            : SAN_JOSE

EMPLOYEE ID                      : SJ0100
FIRST NAME                        : W
MIDDLE INITIAL                   :
LAST NAME                        : WALTERS
WORK DEPT ID                     : F22

PRESS: ENTER TO PROCESS   END TO EXIT

```

Figure 95. Employee Successfully Added

Press the ENTER key to return to the selection panel or END to exit.

Erasing an Employee at a Remote Location

To erase an employee at a remote location, enter the following values:

- On the ACTION line, enter E for erase.
- On the OBJECT line, enter EM for employee.
- On the SEARCH CRITERIA line, enter EI for employee number.
- On the LOCATION line, enter *your server location* for the remote location.
- On the DATA line, enter % which enables you to display a list of all the employees.

```

                                DB2 ORGANIZATION APPLICATION
====>_

ACTION .....:e      A (ADD)          E (ERASE)
                   D (DISPLAY)       U (UPDATE)

OBJECT .....:em     DE (DEPARTMENT)   EM (EMPLOYEE)
                   DS (DEPT STRUCTURE)

SEARCH CRITERIA :ei  DI (DEPARTMENT ID) MN (MANAGER NAME)
                   DN (DEPARTMENT NAME) EI (EMPLOYEE ID)
                   MI (MANAGER ID)     EN (EMPLOYEE NAME)

LOCATION .....:your server location (Blank implies local location)

DATA .....:%

PRESS: ENTER to process   END to exit

```

Figure 96. Starting the Distributed Organization Application

Press the ENTER key. The panel below lists the employees that can be erased. Select the employee to be erased by putting an S in the left margin by the employee ID.

```

                                DB2 ORGANIZATION APPLICATION          ROW 1 OF 4
====>_

ACTION: ERASE  AN EMPLOYEE                      LOCATION: SAN_JOSE

TO SELECT FROM THE LIST PLACE AN S NEXT TO THE EMPLOYEE
PRESS ENTER TO PROCESS OR END TO EXIT

E/ID      EMPLOYEE NAME                      D/ID      DEPARTMENT NAME
S SJ0100   W WALTERS                          F22       BRANCH OFFICE F22
SJ0020   S O'SHEA                          F22       BRANCH OFFICE F22
SJ0030   D COOPER                          F22       BRANCH OFFICE F22
SJ0040   A HAYES                          F22       BRANCH OFFICE F22
SJ0050   L ASHER                          F22       BRANCH OFFICE F22

```

Figure 97. Selecting an Employee at a Remote Location

Press the ENTER key. The panel below displays the information relevant to the selected employee that is to be erased.

```

                                DB2 ORGANIZATION APPLICATION
====>_

                                ERASE  AN EMPLOYEE

DEPARTMENT ID                   : F22
NAME                            : BRANCH OFFICE F22
MANAGER ID                      :
ADMIN DEP ID                    : E01
LOCATION                          : SAN_JOSE

EMPLOYEE ID                     : SJ0100
FIRST NAME                      : W
MIDDLE INITIAL                  :
LAST NAME                       : WALTERS
WORK DEPT ID                    : F22

PRESS: ENTER TO PROCESS  END TO EXIT

```

Figure 98. Employee to be Erased

Press the ENTER key to erase the employee information. A message appears on the panel stating the employee has been successfully erased. You can now erase another employee or press ENTER or END to exit.

```

                                DB2 ORGANIZATION APPLICATION          ROW 1 OF 3
====>
DSN8003I  DSN8HC3-EMPLOYEE SUCCESSFULLY ERASED
ACTION:   ERASE AN EMPLOYEE                                LOCATION: SAN_JOSE

TO SELECT FROM THE LIST PLACE AN S NEXT TO THE EMPLOYEE
PRESS ENTER TO PROCESS OR END TO EXIT

E/ID      EMPLOYEE NAME                                D/ID      DEPARTMENT NAME
SJ0020    S  O'SHEA                                    F22       BRANCH OFFICE F22
SJ0030    D  COOPER                                    F22       BRANCH OFFICE F22
SJ0050    L  ASHER                                     F22       BRANCH OFFICE F22

```

Figure 99. Employee at a Remote Location Erased

Edit Exit Routine

The edit exit routine is prepared in Phase 1. It works with the employee table (DSN8510.EMP) and is written in assembler language.

The name of the edit exit routine is DSN8EAE1. When the employee table (DSN8510.EMP) is changed by either an update or an add, the edit exit routine encodes the salary amount that goes into the SALARY column. When the SALARY column is read from the employee table, the amount is decoded. The encoding and decoding of the salary column protects the confidentiality of the employee's salary.

Huffman Compression Exit Routine

IBM supplies a sample edit routine that compresses data using the Huffman algorithm (first described in *Proceedings of the IRE* September, 1952). Before using any data compression routine, understand its limitations and consider tailoring it to your particular table. For the restrictions and concerns that apply to the IBM sample, see the comments provided with the code. The routine is called DSN8HUFF and resides in library *prefix.SDSNSAMP*.

Sample Field Procedure

A sample field procedure is prepared in Phase 1. This procedure causes values in a CHAR(6) column to be ordered in the ASCII sorting sequence.

Dynamic SQL Statements (DSNTESA, DSNTESQ)

prefix.SDSNSAMP library members DSNTESA and DSNTESQ contain dynamic SQL statements to help verify the success of an installation or migration.

DSNTESA

The SQL statements in DSNTESA are run dynamically by SPUFI or in batch using the sample dynamic SQL program DSNTDP2 and job DSNTDP3P. DSNTESA is used in Phase 3 of the verification process.

The first group of statements in DSNTESA create a temporary table space and table. The INSERT statements fill the table with names, midterm scores, and final examination results, and the SELECT statement then does a check of the averages. The UPDATE statements assign a grade according to the formula in the first UPDATE statement: 60% for the final and 40% for the midterm. The next SELECT statement produces the entire table. The ROLLBACK statement removes the table space and the table within it.

General-use Programming Interface

The following statements make some administrative queries on the system tables:

- The following SELECT statements find all the plans and packages that are owned by the current user, and the date they were bound.

```
SELECT NAME, BINDDATE
       FROM SYSIBM.SYSPLAN
       WHERE CREATOR = USER;
```

```
SELECT COLLID, NAME, VERSION, BINDTIME
       FROM SYSIBM.SYSPACKAGE
       WHERE OWNER = USER;
```

- The following SELECT statements find the plans and packages that require a bind or rebind before they can be run, and the plans and packages that are automatically rebound the next time they are run.

```
SELECT NAME, CREATOR, BINDDATE, VALID, OPERATIVE
       FROM SYSIBM.SYSPLAN
       WHERE OPERATIVE = 'N' OR VALID = 'N';
```

```
SELECT COLLID, NAME, VERSION, BINDTIME, VALID
       FROM SYSIBM.SYSPACKAGE
       WHERE OPERATIVE = 'N' OR VALID = 'N';
```

- The following SELECT statements find all objects required for the current user's programs.

```
SELECT DNAME, BTYPE, BCREATOR, BNAME
       FROM SYSIBM.SYSPLANDEP
       WHERE BCREATOR = USER
       ORDER BY DNAME, BTYPE, BCREATOR, BNAME;
```

```
SELECT DCOLLID, DNAME, BTYPE, BQUALIFIER, BNAME
       FROM SYSIBM.SYSPACKDEP
       WHERE BQUALIFIER = USER
       ORDER BY DCOLLID, DNAME, BTYPE, BQUALIFIER, BNAME;
```

- The second SELECT from SYSTABLES provides information about all the DEPT tables regardless of the owner.

```
SELECT *
       FROM SYSIBM.SYSTABLES
       WHERE NAME = 'DEPT';
```

- The SELECT from SYSCOLUMNS supplies a description of the fields of the DSN8510.DEPT table. This information can also be provided by DCLGEN, and, within a program, the DESCRIBE statement gives this same information.

```
#
#
#
#
SELECT NAME, COLTYPE, LENGTH, SCALE, NULLS, REMARKS, COLNO
FROM SYSIBM.SYSCOLUMNS
WHERE TBNAME= 'DEPT' AND TBCREATOR = 'DSN8510'
ORDER BY COLNO;
```

- The following SELECT statements find the kinds of authority a user can have. Determining which tables a specific user can access is relatively complicated because of the various authorities. If the user has SYSADM authority, any table can be accessed.

```
SELECT * FROM SYSIBM.SYSPLANAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSPACKAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSUSERAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSDBAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSTABAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSCOLAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSRESAUTH WHERE GRANTEE = USER;
```

- The final four SELECT statements show the tables and views that can be accessed directly by the current user, those that can be accessed using a plan, and those that are accessed using the database authority.

```
SELECT TCREATOR, TTNAME, STNAME, GRANTOR
FROM SYSIBM.SYSTABAUTH
WHERE GRANTEE = USER;
SELECT BNAME, BTYPE, GRANTOR, NAME
FROM SYSIBM.SYSPLANAUTH, SYSIBM.SYSPLANDEP
WHERE GRANTEE = USER
AND NAME = DNAME
AND EXECUTEAUTH ^= ' ';
AND (BTYPE = 'T' OR BTYPE = 'V');
SELECT DCOLLID, BNAME, BTYPE, BQUALIFIER, BNAME
FROM SYSIBM.SYSPACKAUTH, SYSIBM.SYSPACKDEP
WHERE GRANTEE = USER
AND COLLID = DCOLLID
AND NAME = DNAME
AND EXECUTEAUTH ^= ' ';
AND (BTYPE = 'T' OR BTYPE = 'V');
SELECT NAME, CREATOR, TYPE, DBNAME, TSNAME
FROM SYSIBM.SYSTABLES
WHERE DBNAME IN
(SELECT NAME FROM SYSIBM.SYSDBAUTH
WHERE GRANTEE = USER
AND DBADMAUTH ^= ' ');
```

_____ End of General-use Programming Interface _____

DSNTESQ

DSNTESQ contains a set of queries to check consistency between catalog tables. The SQL statements are in a format available for input to SPUFI and DSNTDP2. If SPUFI is not bound when you want to execute these queries, you can use the DB2 for MVS/ESA Version 4 DSNTDP2.

Before running these queries, you should run the DSN1CHKR utility to make sure the physical structure of the catalog is correct. You should also run the CHECK INDEX utility.

DSNTESQ contains SQL that creates copies of the catalog using segmented table spaces. In some cases, the queries in DSNTESQ run faster when run on copies of the catalog instead of the actual catalog because the copies have additional indexes. If you plan to use the copies of the catalog, use the comment lines in DSNTESQ for guidance.

A new SELECT statement is added to this sample to identify views that might have problems if they are re-created in DB2 Version 5 because of changes in the WITH CHECK OPTION clause as explained in "SQL92 Entry Level Conformance" on page 2-293.

Dynamic SQL Programs (DSNTIAD, DSNTPE2, DSNTIAUL)

SPUFI is a part of the distributed product. An install job is used to bind it. It can be used only with ISPF. DSNTIAD, DSNTPE2, and DSNTIAUL are sample programs and must be compiled, link-edited, and bound as usual. Authority is required to execute the plan associated with each of these programs:

Program	Plan
DSNTIAD	DSNTIA51
DSNTPE2	DSNTPE51
DSNTIAUL	DSNTIB51

PL/I customers may use DSNTPE2 to issue dynamic SQL in batch mode.

SPUFI, DSNTIAD, DSNTPE2, and DSNTIAUL use dynamic SQL facilities to execute SQL statements. DSNTIAD, DSNTPE2, and DSNTIAUL read statements into the program. They have many similarities, but they also have some significant differences. Input is received from the SYSIN input data set, and the results are sent to the SYSPRINT output data set for these programs. Table 103 summarizes how SPUFI, DSNTIAD, DSNTPE2, and DSNTIAUL handle SQL statements.

Table 103 (Page 1 of 2). SQL Statements Allowed by SPUFI, DSNTIAD, DSNTPE2, and DSNTIAUL

Feature	SPUFI	DSNTIAD	DSNTPE2	DSNTIAUL
SELECT	yes	no	yes	yes ¹
DELETE/INSERT/UPDATE	yes	yes	yes	yes
ALTER/CREATE/DROP	yes	yes	yes	yes
EXPLAIN	yes	yes	yes	yes
GRANT/REVOKE	yes	yes	yes	yes
COMMENT ON	yes	yes	yes	yes
LABEL ON	yes	yes	yes	yes
COMMIT/ROLLBACK	yes	yes	yes	yes
LOCK TABLE	yes	yes	yes	yes
CONNECT	no ²	yes	yes	yes
SET CONNECTION	no	yes	yes	yes

Table 103 (Page 2 of 2). SQL Statements Allowed by SPUFI, DSNTIAD, DSNTPE2, and DSNTIAUL

Feature	SPUFI	DSNTIAD	DSNTPE2	DSNTIAUL
RELEASE	no	yes	yes	yes
-- comment	yes	no	yes	no
* comment	no	no	yes	no
SQL error	ROLLBACK	Continue (return code 8)	Continue (return code 8)	Continue (return code 8)

Note:

¹ A SELECT statement is used by DSNTIAUL to determine the tables to be unloaded. See “DSNTIAUL Sample Unload Program” on page 2-368 for more information.

² You can connect to a remote location by specifying a remote location name in the first SPUFI panel.

DSNTIAD Sample Program

DSNTIAD is a DB2 sample program written in assembler that uses dynamic SQL. It can be used to issue all SQL statements except SELECT in batch mode. Job DSNTIJTM prepares and then uses program DSNTIAD for the installation and migration process.

The input and output data sets used by this sample program are as follows:

Data Set	Description
SYSIN	SQL statement input. Only the first 72 columns are considered.
SYSPRINT	Output listing.

DSNTIAD accepts all SQL statements that can be dynamically prepared as input, with the exception of SELECT. If DSNTIAD is bound into an application package, it also accepts the CONNECT, SET CONNECTION, and RELEASE statements. DSNTIAD does not accept delimited location names in CONNECT, SET CONNECTION, and RELEASE. DSNTIAD concatenates input statements until a semicolon is encountered or until the input buffer overflows. Because DSNTIAD uses a simple scanning algorithm that makes no allowance for string delimiters, semicolons are not permitted within the syntax of input statements. DSNTIAD does not support any comment statements.

It is possible to force DSNTIAD to complete with a return code of 0 by specifying PARM('RC0') at execution time.

Special Considerations: DSNTIAD can process SQL in batch mode only, but there are some differences between SPUFI and DSNTIAD. In case of error, SPUFI issues a ROLLBACK to the last COMMIT. DSNTIAD continues executing the next input statement, allowing up to 10 errors. When the input statement is finished, a return code of 8 is issued. It also terminates after some SQL errors, but it does not issue a ROLLBACK for error situations. For more details, see the source in *prefix.SDSNSAMP*, or examine the program comments in the assembler listing.

Listing: For an example of the JCL required to prepare and execute DSNTIAD, refer to the output you receive from running job DSNTIJTM. Job DSNTEJ1, which is part of Phase 1, also executes program DSNTIAD.

DSNTEP2 Sample Program

DSNTEP2 is a sample program written in PL/I and provided in source form. It runs SQL statements in batch mode. Program DSNTEP2 handles both DDL and DML SQL statements, including SELECT. In addition, support is provided for the static SQL statements of CONNECT, SET CONNECTION, and RELEASE. However, DSNTEP2 does not support delimited location names in the SQL statements CONNECT, SET CONNECTION, and RELEASE. DSNTEP2 accepts a "*" in column 1 or "--" anywhere on a line to indicate a comment. However, you cannot embed comments anywhere within an SQL statement.

The input and output data sets used by this sample program are as follows:

Data Set	Description
SYSIN	SQL statement input. Only the first 72 columns are considered.
SYSPRINT	Output listing.

Special Considerations: DSNTEP2 provides a batch mode SQL processing capability, but there are some differences between SPUFI and DSNTEP2. In case of an error, SPUFI issues a ROLLBACK to the last commit. DSNTEP2 allows up to 10 errors (MAXERRORS) and continues executing the next input statement, finishing with a return code of 8. It also terminates after some SQL errors, but it does not issue a ROLLBACK for error situations. For more details, see the source in *prefix.SDSNSAMP*, or examine the program comments in the compiler listing.

DSNTEP2 supports identifiers and comments that include double-byte characters. Specify the program input parameter MIXED for the characters x'0E' and x'0F' to mean DBCS shift-in and shift-out characters. The default parameter is NOMIXED, which gives no special meaning to these characters.

Output Format: By default, DSNTEP2 centers the output from SELECT statements on the page. You can change the alignment with these run options:

PARMS ALIGN(LHS) Align the output on the left-hand side
PARMS ALIGN(MID) Center the output

Listing: A listing of program DSNTEP2 can be obtained by compiling it, as done in job DSNTEJ1P. The source can be found in *prefix.SDSNSAMP*. An example of DSNTEP2 execution is provided in job DSNTEJ1P. The expected results from that job are shown in "Appendix A. Sample Application Output" on page X-3.

DSNTIAUL Sample Unload Program

DSNTIAUL is an assembler language sample program that unloads tables into sequential data sets that can later be reloaded into the same DB2 table or into another DB2 table. The source can be found in *prefix.SDSNSAMP*.

Program DSNTIAUL is prepared and run in Phase 2 by job DSNTEJ2A. When you run this program, you can specify the 'SQL' parameter. This allows you to enter complete SQL statements; each statement must end with a semi-colon. DSNTIAUL uses SQL SELECT statements to determine which tables and columns to unload. You can enter any other statements that can be executed dynamically. For example, use the LOCK TABLE statement to lock a table before you unload it. In addition, if you bind DSNTIAUL into an application package, you can enter the CONNECT, SET CONNECTION, and RELEASE statements to access data at remote locations.

The sample unload program allows you to:

- Unload DB2 data from tables and views
- Provide instructions to the subsystem for loading DB2 data
- Move or rearrange data objects within and between DB2 subsystems
- Enter SQL statements (if you have invoked DSNTIAUL with the SQL option).

Some examples of ways you can use the sample unload program are to:

- Transport data within and between subsystems. You can test the same application on a variety of DB2 subsystems to check results for consistency.
- Duplicate data. To test applications on realistic data, you can make backup copies of the data for testing purposes, thus preventing alteration or destruction of the original data.
- Restructure data. With the unload program, you can add or delete rows and columns to or from tables and views, join separate tables and views, and unload and reload views from and into tables.

Unload Processing: The input and output data sets used by the unload program are as follows:

Data Set	Description
-----------------	--------------------

SYSIN	Input data set. There are two formats for data in this data set.
-------	--

#

- Complete SQL statements that end with a semi-colon. These are only valid if you specified PARM('SQL') when you invoked DSNTIAUL. You can use any number of SQL statements. Each SQL statement can be up to 32765 bytes long and be continued on multiple records. However, each record must be no more than 72 bytes long. SELECT statements are used to unload tables, and non-SELECT statements are executed dynamically. If you select data from more than one table or you select only a subset of columns from a table, you need to edit the LOAD statement generated by DSNTIAUL. The LOAD statement will have a generated table name and might be missing some column names.

#

- Single-line statements (without a semi-colon) that use the following syntax:

table/view name [WHERE conditions] [ORDER BY columns]

Each input line should look like a valid SQL SELECT statement, with the SELECT * FROM clause omitted. It must be no more than 72 bytes long and must not be continued on multiple lines.

SYSPRINT	Output data set. It contains any messages generated during the execution of the program.
----------	--

SYSPUNCH	Output data set. It contains the LOAD utility control statements to load the tables. You can specify the record length and block size of this data set.
----------	---

SYSRECnn	Output data sets. The value <i>nn</i> ranges from 00 to 99. Each data set contains the data unloaded from one of the objects specified in SYSIN. The maximum record length for an unload data set is 32760 bytes.
----------	---

Data control block (DCB) information for the SYSPRINT data set is in the program and must not be coded in the JCL. You can code the record length and block size information for SYSPUNCH and SYSRECnn data sets.

The unload process, using the data sets, enables you to accomplish the following work:

Move Data: To move data, proceed as follows:

1. Unload the desired tables using DSNTIAUL.
2. Create new or revised tables on a different subsystem.
3. Reload the tables using data from the SYSRECnn data set and load control statements (created by the UNLOAD program) from the SYSPUNCH data set.

Duplicate Tables: To duplicate tables, proceed as follows:

1. Unload the desired tables using DSNTIAUL.
2. Create tables with different names.
3. Edit the LOAD utility control statements contained in the SYSPUNCH data set to indicate new table names.
4. Reload the tables using the data from the SYSRECnn data sets and the load control statements from the edited SYSPUNCH data set.

Restructure Tables: To restructure tables, proceed as follows:

1. Create the desired views with names of tables as the targets.
2. Unload the desired views.
3. Create new, restructured tables.
4. Edit the LOAD utility control statements contained in the SYSPUNCH data set if required.
5. Reload the new tables using the data from the SYSRECnn data sets and the load control statements from the SYSPUNCH data set.

UNLOAD Listing: An example of DSNTIAUL execution is provided in job DSNTJ2A. The expected results from that job are shown in "Appendix A. Sample Application Output" on page X-3.

Chapter 2-10. Connecting the IMS Attachment Facility

This chapter covers the requirements for connecting the IMS attachment facility from a DB2 perspective and refers you to IMS books for specific IMS information. Connecting DB2 to IMS requires coordination with your IMS support group.

To connect the IMS attachment facility, you must:

- Make DB2 load modules available to IMS
- Define DB2 to IMS
- Define new programs and transactions to IMS.

Depending on your site, you might also need to:

- Define DB2 plans for IMS applications
- Generate a user language interface.

These tasks are discussed below. An IMS system definition might be required to perform these steps. If RACF is installed, you also need to define the IMS-to-DB2 connection to RACF. For information on how to do this, see Section 3 (Volume 1) of *Administration Guide*.

Making DB2 Load Modules Available to IMS

If you have already included the *prefix.SDSNLOAD* library in your LNKSTxx, you can skip this step. Version 5 modules will be available through normal MVS module search.

Connecting to more than one release of DB2: If any IMS region connects to more than one release of DB2, then you must ensure that the DB2 load library used for that region is compatible with each release. The IMS attachment facility is upward compatible, but not downward compatible. This means you should use the oldest release of the DB2 load library for the IMS region.

If you have not included the DB2 load libraries in your LNKSTxx, you must add STEPLIB statements to your startup procedures and add *prefix.SDSNLOAD* to the DFSESL DD statement.

- If all the data sets referred to in the JOBLIB or STEPLIB statement for an IMS region are APF-authorized, then add the DD statement for *prefix.SDSNLOAD* to the JOBLIB or STEPLIB statement. If the DYNAM option of COBOL II is being used, the IMS RESLIB DD statement must precede the reference to *prefix.SDSNLOAD* in the JOBLIB or STEPLIB statement.
- Add the ddname DFSESL DD statement for *prefix.SDSNLOAD*. All libraries specified on the DFSESL DD statement must be APF-authorized. The DFSESL DD statement is not required by DB2 DL/I batch support. IMS requires that an IMS RESLIB DD statement also be referenced by the DFSESL DD statement, as in the following:

```
//DFSESL DD DSN=ims_reslib,DISP=SHR
// DD DSN=prefix.SDSNLOAD,DISP=SHR
```

Defining DB2 to IMS

The DB2 identification must be defined to the control region, the DL/I batch region, and, optionally, to each dependent region accessing that DB2 system. To make this identification, you must create a subsystem member (SSM) in the IMS.PROCLIB library, and identify the SSM to the applicable IMS regions.

The DB2 identification for DL/I batch has more parameters than the control and dependent regions. For information on DL/I batch, see Section 4 of *Application Programming and SQL Guide*.

Placing the Subsystem Member Entry in IMS.PROCLIB: Each SSM entry in IMS.PROCLIB defines at least one connection from an IMS region to at least one different MVS subsystem.

To name an SSM member, concatenate the value (one to four alphanumeric characters) of the IMSID field of the IMS IMSCTRL macro with any name (one to four alphanumeric characters) defined by your site.

One SSM member can be shared by all of the IMS regions, or a specific member can be defined for each region. This record contains as many entries as there are connections to external subsystems. Each entry is an 80-character blocked or deblocked record. The following examples show how to define fields for IMS. Fields are keyword or positional and are delimited by commas. For more information, see *IMS/ESA Customization Guide* from the appropriate release. The fields in this record are:

SST=,SSN=,LIT=,ESMT=,RTT=,REO=,CRC=

where:

SST=DB2

is a required one-to eight-character name which defines the external subsystem type. It must be set to DB2 for IMS to connect to DB2.

SSN=

is a required one-to four-character DB2 subsystem name. This name must be the name you specified for SUBSYSTEM NAME on installation panel DSNTIPM. The default is DSN1.

LIT=

is a required four-character alphanumeric option, specifying the language interface token (LIT) supplied to IMS. The IMS-supplied language interface module (DFSLI000) requires a value of **SYS1** for this option.

If you need to define connections to different DB2 subsystems, you can follow the procedure described in “Defining DB2 Plans for IMS Applications (Optional)” on page 2-376.

ESMT=

is a required one-to eight-character alphanumeric option specifying the external subsystem module table. This module specifies which attachment modules must be loaded by IMS. **DSNMIN10** is the required value for this field.

RTT=

is an optional one to eight character alphanumeric name of the user-generated resource translation table (RTT). This table maps the IMS application names

into DB2 plan names. If this entry is omitted, the DB2 plan name is the IMS application load module name.

See “Defining DB2 Plans for IMS Applications (Optional)” on page 2-376 for details on how to generate a resource translation table.

REO=

is the optional one-character region error option to be used if an IMS application attempts to reference a non-operational external subsystem or if resources are unavailable at create thread time. If DB2 detects the unavailable resource condition during normal SQL processing, a -904 SQLCODE is returned to the application.

R passes a SQL return code to the application, indicating that the request for DB2 services failed (default). The most commonly returned SQL codes are -922, -923, and -924. However, there might be other SQL codes returned to the application.

When the first connection to DB2 cannot be established, a SQL return code is not returned. Instead, the application is terminated with an abend code U3047.

Q abends the application with an abend code U3051, backs out activity to the last commit point, does a PSTOP of the transaction, and requeues the input message. This option only applies when an IMS application attempts to reference a non-operational external subsystem or if the resources are unavailable at create thread time. If DB2 detects the unavailable resource condition during normal SQL processing, a -904 SQLCODE is returned to the application.

A abends the application with an abend code of U3047 and discards the input message. This option only applies when an IMS application attempts to reference a non-operational external subsystem or if the resources are unavailable at create thread time. If DB2 detects the unavailable resource condition during normal SQL processing, a -904 SQLCODE is returned to the application.

If DB2 is not active or the connection cannot be established when the first SQL call is made from the application program (such as DB2 unavailable, DB2 quiescing, or DB2 terminating), the action you take depends on the region error option specified. SQL codes of -922, -923, or -924 might be returned to the application if option R is specified.

You can change the default for an application if a resource translation table entry is generated for that application. See “Defining DB2 Plans for IMS Applications (Optional)” on page 2-376.

CRC=

is a command recognition character used by IMS to identify DB2 commands entered from an IMS terminal with the /SSR command. Any character is valid for the CRC except the period (.), slash (/), or comma (,). The default CRC is the hyphen (-).

These options apply to DL/I batch only:

CONNECTION_NAME=

The connection name is optional. It represents the name of the job step that is the coordinator for DB2 activity. The connection name defaults are:

Table 104. Default Connection Names for DL/I Batch

Type of Application	Default Connection Name
Batch job	Job name
Started task	Started task name
TSO user	TSO authorization ID

If a batch job fails, you must use a separate job to restart the batch job. The connection name used in the restart job must be the same as the name used in the batch job that failed. Or, if the default connection name is used, the restart job must have the same job name as the batch update job that failed.

DB2 requires unique connection names for DB2 DL/I batch support. If two applications try to connect with the same connection name, then the second application is not allowed to connect to DB2. CONNECTION_NAME can be 1-8 characters long.

PLAN=

You can specify a DB2 plan name. If you do not specify a plan name, the application program module name is checked against the optional resource translation table. If a match is found, the translated name is used as the DB2 plan name. If no match is found, the application program module name is used as the plan name. PLAN can be 1-8 characters long.

PROG=

You must specify the name of the application program to be loaded and to receive control. PROG can be 1-8 characters long.

Providing IMS Support for DB2 Commands: You can enter DB2 commands through the /SSR command of IMS. The /SSR command format is:

/SSR crc DB2 command

as in

/SSR -DISPLAY THREAD ()*

IMS supports this command; you must define the CRC in the SSM member of the IMS control region. If the /SSR command is entered through the MVS console, the AUTHID WTOR needs to be granted the appropriate authority. If the /SSR command is entered through an IMS terminal, the IMS LTERM name or the signon ID (if active) needs to be granted the appropriate authority.

Specifying the SSM EXEC Parameter: Specify the SSM EXEC parameter in the startup procedure of the IMS control, MPP, BMP, or DL/I batch region. The SSM is concatenated with the IMSID to form a member name in IMS.PROCLIB. The IMSID comes from the IMSID option of the IMSCTRL generation macro or the IMSID option in the control region startup procedure.

For DL/I batch regions, you can specify the DB2 connection parameters in the DDITV02 data set instead of an SSM member. The DDITV02 data set and an SSM member have the same format. See Section 4 of *Application Programming and SQL Guide* for more information on the DDITV02 data set.

If you specify the SSM for the IMS control region, any dependent region running under the control region can attach to the DB2 subsystem named in the IMS.PROCLIB member specified by the SSM parameter. The IMS.PROCLIB

member name is the IMS ID (IMSID=xxxx) concatenated with the one to four characters specified in the SSM EXEC parameter. The IMS ID is the IMSID parameter of the IMSCTRL generation macro.

IMS allows you to define as many external subsystem connections as are required. More than one connection can be defined for different DB2 subsystems. All DB2 connections must be within the same MVS system. For a dependent region, you can specify a dependent region SSM or use the one specified for the control region. You can specify different region error options (REOs) in the dependent region SSM member and the control region SSM member. Table 105 shows the different possibilities of SSM specifications.

Table 105. SSM Specifications Options

SSM for Control Region	SSM for Dependent Region	Action	Comments
No	No	None	No external subsystem can be connected.
No	Yes	None	No external subsystem can be connected.
Yes	No	Use the control region SSM.	Applications scheduled in the region can access external subsystems identified in the control region SSM. Exits and control blocks for each attachment are loaded into the control region address space.
Yes	Yes (NULL entry)	No SSM is used for the dependent region.	Applications scheduled in this region can access DL/I databases only. Exits and control blocks for each attachment are loaded into the control region address space and each dependent region address space.
Yes	Yes	Check the dependent region SSM with the control region SSM.	Applications scheduled in this region can access only external subsystems identified in both SSMs. Exits and control blocks for each attachment are loaded into the control region and the dependent region address space.

No specific parameter exists to control the maximum number of SSM specification possibilities.

Defining New Programs and Transactions to IMS

Programs and transactions already defined to IMS can use SQL without any additional definition to IMS.

You can define new programs and transactions that access DB2 resources to your IMS system. Coordinate with your IMS support group to install the programs and transactions for Phase 4 of the verification process. For more information, see “Chapter 2-9. Verifying with the Sample Applications” on page 2-305.

Defining DB2 Plans for IMS Applications (Optional)

The application plan defines the DB2 resources being accessed from an application. The application plan is identified by its plan name. Each IMS application is associated with a plan name.

The default is to have the DB2 plan name the same as the IMS application program load module name. We recommend that you use the default.

If you assigned a different name to the plan, you need a resource translation table (RTT). If you chose an error option different from the REO default, you also need an RTT. DB2 provides the DSNMAPN macro in *prefix.SDSNMACS* to generate an RTT. After it is assembled, the table must be link-edited as REENTRANT with RMODE=24 into any authorized library that is concatenated with the library from which IMS loads the DB2 IMS attach modules.

The format of DSNMAPN macro is shown in Table 106.

Table 106. DSNMAPN Macro Format

Macro	Option	Meaning
DSNMAPN	APN=	IMS application name
	,PLAN=	Associated DB2 plan name
	[,OPTION=]	Specific entry error option R, Q, or A. See REO in the SSM entry.
	[,END=]	Indicates last entry (YES/NO). NO is the default.

This macro is described in more detail in “IMS Attachment Facility Macro (DSNMAPN)” on page 2-377.

Generating a User Language Interface (Optional)

This step is required only if you intend to access two DB2 subsystems from the same dependent region.

To provide this access, the SSM must contain one entry for each subsystem. Each entry contains a different subsystem ID and its associated language interface token (LIT). IMS provides the DFSLI macro to generate additional language interface modules with unique LITs. The general format of the macro is shown in Table 107.

Table 107. DFSLI Macro Format and Meaning

Macro	Option	Meaning
DFSLI	TYPE	Specifies the type of subsystem that can be accessed through this language interface module. DB2 is the only value supported by this option
	LIT	Defines a name (called LIT) to relate a language interface module with an entry in the SSM for the dependent region

When an IMS application issues a DB2 request, IMS knows the target subsystem by the LIT used in the request. For example, consider the case of a dependent region accessing two DB2 subsystems (DSN1 and DSN2):

- You generate a language interface with LIT=SYS2 (DFSLI001).
- You define two entries in the SSM member. The first entry points to DSN1 with LIT=SYS1; the second points to DSN2 with LIT=SYS2.
- You link-edit applications accessing the DSN1 subsystem with the IMS-provided language interface (DFSLI000).
- You link-edit applications accessing the DSN2 subsystem with the user-generated language interface (DFSLI001).

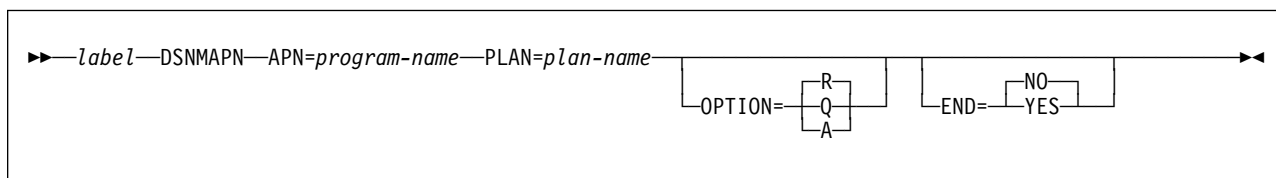
Even though a region can communicate with two or more DB2 subsystems, an IMS application can access only one—the DB2 subsystem referred to in the language interface that is link-edited. You can alter the SSM to route application requests to a different DB2 subsystem.

IMS Attachment Facility Macro (DSNMAPN)

This macro is required only when an IMS application load module name is different from the name of its related IBM DATABASE 2 application plan, or if the error option is different from the ERR value specified on the IMS SSM entry.

Macro statements are assembled in *prefix*.SDSNMACS and must be link-edited as REENTRANT with RMODE=24 into the DB2 library *prefix*.SDSNLOAD. The module name must be specified on the IMS SSM entry for the DB2 subsystem. The name must be specified as in the RTT entry for the SSM member defining the connection of this region. IMS loads the RTT module into the dependent region address space.

For more information on the RTT, refer to “Defining DB2 Plans for IMS Applications (Optional)” on page 2-376.



Notes:

1. The macro name must be followed by one or more blanks before options are coded.
2. Multiple options must be separated by commas (with no blanks).

Description

label DSNMAPN

DSNMAPN is the name of the macro. It must be coded exactly as it appears here, and it must be separated from any optional options by one or more blanks.

For *label*, substitute the CSECT name of your module. This name must match the name of the module specified to the linkage editor. *Label* is optional except

for the first invocation of the DSNMAPN macro. The last invocation requires END=YES.

APN=*program-name*

Specifies the name of an application load module scheduled by IMS. For *program-name*, substitute an application name of up to eight characters.

PLAN=*plan-name*

Specifies an application plan name that is used (instead of the default application name) when a thread is created. For *plan-name*, substitute an application plan name of up to eight characters.

OPTION=R|Q|A

Specifies the action taken when an application program call cannot be performed because there is some problem in communication between the application program and the DB2 subsystem or if resources are unavailable.

If OPTION is not specified, the region error option (REO) is used.

- R** Specifies that a return code is returned to the application to indicate that the request for DB2 services failed.
- Q** Specifies that the transaction is abnormally terminated with an abend code U3051, activity is backed out to the last commit point, and the input message is requeued.
- A** Specifies that the transaction is abended with an abend code of U3047, and the input message is deleted.

Default: R

END=NO|YES

Specifies whether this is the last DSNMAPN macro invocation.

NO

Specifies that this is not the last DSNMAPN macro invocation.

YES

Specifies that this is the last DSNMAPN macro invocation.

Default: NO

The last DSNMAPN macro invocation must be followed by the specification END=YES.

Usage Notes

- To enter more than one application name (with its corresponding plan name and OPTION specification), you must use multiple invocations of the DSNMAPN macro. The first invocation requires the label; the last invocation requires END=YES.
- Invocations must be in ascending order by application name. If they are not, an MNOTE macro error is generated.

Chapter 2-11. Connecting the CICS Attachment Facility

This chapter describes activities required to support CICS in a DB2 environment. Coordinate the connection of the CICS attachment facility with your CICS support group.

To connect the CICS attachment facility, whether installing or migrating, you must do the following:

1. Link CICS routines with DB2 load modules by running job DSNTIJSU. (Not necessary for CICS Version 4.)
2. Calculate space requirements for the CICS attachment facility. Recalculate if you are migrating.
3. Define the DB2 to CICS connection using the RCT. For migration either to DB2 for OS/390 Version 5 or to CICS Version 4, reassemble your RCT.
4. Update the CICS system tables. (Not always necessary for migration.)
5. Update the CICS initialization JCL.
6. Coordinate DB2 and CICS security. (Not always necessary for migration.)
7. Prepare CICS applications for DB2. (Not always necessary for migration.)

If RACF is installed, you also need to define the CICS-to-DB2 connection to RACF. For information on how to do this, see Section 3 (Volume 1) of *Administration Guide*.

Using the Attachment Facility for CICS Version 4 and Subsequent Releases

CICS Version 4 provides a new attachment facility that is shipped on the CICS product tape. You must use the new attachment facility to connect to DB2 from CICS Version 4. You can use the new facility with DB2 Version 2 Release 3 and subsequent releases.

The new attachment facility running under CICS Version 4 has the following enhancements:

- The new attachment facility can run above 16M.
- The new attachment facility has some performance benefits over old attachment facility.
- The new attachment facility traces provide more information.
- You can now easily identify waits for DB2 on CEMT I TAS commands and on DB2 formatted region dumps.
- You can specify a two-byte suffix and three new options for your resource control table (RCT).
- You can use the CICS INITPARM to specify your RCT suffix and DB2 SSID defaults. You can also use the DSNCL STRT command to specify both your RCT suffix and SSID.
- You can run SQL programs in PLT initialization after running DSN2COM0.
- You can recover indoubts when connecting multiple DB2s from CICS.
- You can use the INQUIRE EXITPROGRAM command to help prevent AEY9 abends.

With the new attachment facility, you no longer need to run job DSNTIJSU. CICS Version 4 provides the CICS system definition (CSD) definitions for the attachment facility programs and transactions. If you are sharing the CSD between different releases of CICS, see “Sharing the CSD between CICS/ESA Version 4 and prior releases” in the *CICS/ESA Installation Guide* before deleting program definitions. Check for down-level attachment facility program definitions in your CSD. Delete any of these program definitions if your CSD is for **CICS Version 4 only**:

DSNCCOM0	DSNCCOM1	DSNCCOM2	DSNCEDF1
DSNCEDON	DSNCEXT1	DSNCEXT2	DSNCMSG0
DSNCSTOP	DSNCSTRT	DSNCUEXT	

Delete any of these transaction definitions if your CSD is for CICS Version 4 only:

DSNC
any transaction that executes program DSNCCOM0, DSNCCOM1, or
DSNCCOM2

Most module names for the new attachment facility have been renamed from DSNCxxxx to DSN2xxxx.

Using the Attachment Facility for CICS Version 3

DB2 provides facilities that attach to CICS services for CICS releases earlier than Version 4. To use these facilities, the appropriate CICS routines must be included in the DB2 load modules. These routines are linked by installation job DSNTIJSU.

CICS/ESA Version 3 eliminates the use of the program control table (PCT) and the processing program table (PPT). Instead, CICS/ESA Version 3 requires the use of resource definition online (RDO) to define transaction and program entries. If you use CICS/ESA 3.1 or above, job DSNTIJSU also defines the RDO application program and transaction entries that you need for DB2. You must ensure that the CICS group used in DSNTIJSU is for the exclusive use of DB2-supplied entries, because DSNTIJSU deletes and re-creates the group. In addition, you should ensure that the CICS list you use is added to the list specified in your CICS startup parameters.

If you filled in fields 6, 7, 10, and 11 on installation panel DSNTIP3 (values for these fields may or may not be required, depending on the CICS release you use), the values you specified are edited into the LOADCSD step of job DSNTIJSU. This step executes CICS resource definition online (RDO) statements for the DB2 CICS attachment facility.

If DSNTIJSU runs successfully, it produces return codes of 0.

Once you have run DSNTIJSU for installation or migration, you should not have to run it again, even when applying maintenance to the CICS attachment facility modules. If you get unresolved external references for DFHEAI or DFHEAI0 applications of a PTF, there may be a problem with your target library. DSNTIJSU generally resolves external references for DFHEAI and DFHEAI0 in the CICS attachment facility load modules.

Calculating Space Requirements for the CICS Attachment Facility

The DB2 CICS attachment facility requires space in several portions of the CICS address space. The relevance of each of these areas to the CICS attachment facility is described in this section. Because many storage areas have changed in CICS Version 3 and Version 4, the versions are addressed separately.

For storage estimates for each CICS region, see Table 108 on page 2-382, Table 109 on page 2-382, and Table 110 on page 2-383. For more information on CICS storage considerations, see the appropriate CICS performance guide for your release.

CICS Version 3

- EDSA** Execution Diagnostic Facility (EDF) information is stored in this area, which is the EDSASZE in the CICS system initialization table (SIT).
- MVS Storage** This area is comparable to OSCOR in prior CICS releases. However, the order of storage allocation has changed. This area is what remains in the CICS region after all other areas have been allocated. Because users can no longer specify the size of OSCOR, be sure that the difference between the CICS region size and the DSA size (DSASZE) coded in the CICS system initialization table (SIT) is large enough to satisfy the storage estimates shown in Table 108 on page 2-382, Table 109 on page 2-382, and Table 110 on page 2-383.
- MVS Storage Above Region** This area is comparable to the OS Free Storage Above Region area in prior releases of CICS.

<i>Table 108. Virtual Storage Requested for CICS Region</i>				
Storage Area	Amount	Description	Factor	Notes
OS Free Storage (OSCOR)	0.7KB	DB2 CICS attach control blocks	Per TCB	To obtain the maximum number of TCBs required, use the RCT THRDMAX value. If not coded, the default value is 12 .
"	0.5KB	DB2 CICS Attach control blocks	Per address space	This value varies widely because it is dependent on RCT values for THRDM , THRDA , and TWAIT . See "Estimating Storage Needed for CICS Attachment Threads" on page 2-384 for calculation details.
"	11.2KB	DB2 CICS attach modules	Per address space	
"	5.0KB	Resource control table (RCT)	Per address space	Size varies, depending upon number of entries. For estimating purposes, a 50-entry RCT was assumed. Obtain exact value from RCT link-edit map.
CICS DSA	25.2KB	CICS attach modules	Per address space	
"	2.3KB	DB2 CICS control block	Per active thread	
"	1.1KB	CICS control block	Per address space	
EDSA	12KB	EDF diagnostic screens	Per active thread in EDF mode	CICS Version 3 and above. This storage is freed when EDF mode is deactivated.

<i>Table 109. Contents of CICS Address Space High Private Area</i>				
Storage Area	Amount	Description	Factor	Note
LSQA, SP229, and 230	9.5KB	MVS and DB2 control blocks	Per active thread	
OS Free Storage (Available for Region)	3KB	MVS recovery termination management work areas	Per active thread	If there is not enough free storage for recovery termination management (RTM) processing, the CICS address space may be terminated with a 40D abend.

Storage Category	Per CICS Address Space	Per Thread
CICS OSCOR	11.7KB + RCT size (5.0KB)	0.7KB
CICS Dynamic Storage Area (DSA)	26.3KB	2.3KB
EDSA (CICS Version 3 and above)	0	12.0KB
LSQA, SP229, and 230	0	9.5KB
OS Free Storage	0	3.0KB
Total	43.0KB	27.5KB

CICS Version 4

EDSA Execution Diagnostic Facility (EDF) information is stored in this area, which is the EDSASZE in the CICS system initialization table (SIT).

MVS Storage This area is the CICS region reserved for non-CICS functions such as DB2 and VSAM.

MVS Storage Above Region
This area is comparable to the OS Free Storage Above Region area in prior releases of CICS.

Storage Area	Amount	Description	Factor	Notes
MVS storage	0.7KB	DB2 CICS Attach Control Blocks	Per TCB	Above 16M.
"	0.5KB	DB2 CICS Attach Control Blocks	Per address space.	Varies widely because it is dependent on various RCT settings. See "Estimating Storage Needed for CICS Attachment Threads" on page 2-384 for more information.
"	9.8KB	DB2 CICS Attach MVS Subtask Modules	Per address space	Above 16M.
"	5.1KB	Resource Control Table (RCT)	Per address space	Size varies. Check RCT link-edit map. For estimating purposes, a 50-entry RCT was assumed. Below 16M.
"	0.1KB	GETMAINed Work Area	Per address space	Above 16M.
"	<0.1KB	INDOUBT Parmlist	Per address space	Above 16M.
CICS DSA (ERDSA)	28.1KB	CICS Attach Modules	Per address space	Only resident DB2 CICS Attach modules are considered here since other are used infrequently and occupied storage is freed. Above 16M.
"	0.8KB	CICS Control Block	Per active thread	Above 16M.
"	0.6KB	CICS GETMAINed Storage	Per address space	Above 16M.

Storage Area	Amount	Description	Factor	Notes
LSQA, SP229, and 230	5.5KB	MVS and DB2 Control Blocks	Per active thread	
MVS Free Storage (Available for Region)	3KB	MVS Recovery Termination Management Work Areas	Per active thread	If there is not enough free storage for recovery termination management (RTM) processing, CICS address space may be terminated with a 40D abend.

Storage Category	Per CICS Address Space	Per Thread
MVS Storage	10.0KB + RCT size	0.7KB
CICS Dynamic Storage Area (ERDSA)	28.7KB	0.8KB
MVS Free Storage (Available for Region)	0	3KB
LSQA, SP229, and 230	0	5.5KB
Total	43.8	12.5KB

Estimating Storage Needed for CICS Attachment Threads

When the CICS attachment facility starts, it allocates storage for the maximum number of threads that can be created for each transaction in the RCT. To determine the amount of storage (in bytes) the CICS attachment facility needs to keep track of threads, perform the following calculations for each TYPE=ENTRY, TYPE=POOL, and TYPE=COMD entry in the resource control table. (Remember that the TYPE=COMD and TYPE=POOL macros are provided for you if you do not code them, so be sure to include them in your calculations. Table 114 shows their default values.)

TYPE=	THRDM	THRDA	TWAIT
COMD	1	1	POOL
POOL	3	3	YES

For each item in the RCT, perform the following calculation:

- If THRDM > 0:
 1. Determine the following value:

$$20 \times \text{THRDM} + 44$$
 2. If TWAIT=YES, perform the following calculation, dropping all fractions:

$$\text{THRDM}/4 + (\text{THRDM} - \text{THRDA})/4 + 6$$
 3. If TWAIT=NO or POOL, use the value of THRDM or 3, whichever is smaller.
 4. Multiply the value from either step 2 or step 3 by 4.
 5. Add the results of steps 1 and 4.
- If THRDM=0, use 44.

After performing the preceding calculations on every item defined in the RCT, add 276 bytes to the final calculation. This is amount of storage that is allocated for all threads in the RCT.

Defining the DB2 to CICS Connection Using the RCT

The CICS-to-DB2 connection is defined by creating and assembling the resource control table (RCT). This table describes the relationship between CICS transactions and DB2 resources. The DB2 attachment facility provides a macro to generate the RCT. For CICS Version 4, you must use the DSNCRCT macro that is shipped with CICS410.SDFHMAC instead of the one in your DB2 macro library. For CICS 3.3 and earlier, you must use the DSNCRCT macro provided in *prefix.SDSNMACS*. For a complete description of this macro and how to assemble it, see “CICS Attachment Facility Macro (DSNCRCT)” on page 2-398. For information on specifying the RCT suffix, see page 2-396.

Plan for performance before connecting DB2 to CICS because some of the parameters influence performance. See Section 5 (Volume 2) of *Administration Guide* for more information.

Using the CICS attachment facility macro, you define the following:

- RCT entries for a transaction or a group of transactions and their corresponding transaction identifiers
- Plan names
- Authorization checking options
- Thread subtask limits
- Type of processing to follow create thread errors.

To generate the RCT, specify:

- One TYPE=INIT macro
- An optional TYPE=COMD macro, or assume the default command entry
- An optional TYPE=POOL macro, or assume the default pool entry
- As many TYPE=ENTRY macros as required
- One TYPE=FINAL macro.

The RCT must be link-edited into a library that is accessible to MVS through its normal library search order (STEPLIB, JOBLIB, link library). Do not link the RCT as REENTRANT or REUSABLE because this causes the CICS attachment facility to abend on startup. You must link the RCT as RMODE(24).

The entries needed in the RCT to run Phase 5 of the installation verification procedure are listed in *prefix.SDSNSAMP(DSN8FRCT)*. For more information on installation verification, see “Phase 5: Testing the CICS Environment” on page 2-327.

Updating the CICS System Tables

Connecting DB2 to CICS requires that you regenerate several CICS tables with additional entries. You can use the resource definition online feature of CICS to add the appropriate transaction and program entries.

The attachment facility programs have been renamed for the CICS Version 4. In this section, we use *y* as the fourth character of the program names listed in Table 115 on page 2-386 (for example, DSN y COM0). For CICS Version 3 and earlier, *y* is C. For CICS Version 4, *y* is 2.

Table 115. Attachment Facility Program Names

CICS Version 3 and Earlier	CICS Version 4
DSNCCOM0	DSN2COM0
DSNCCOM1	DSN2COM1
DSNCCOM2	DSN2COM2

System Initialization Table Entries

- Program list table (PLT) processing (optional)

When using PLT processing, enter the PLT suffixes to the PLTPI entry, the PLTSD entry, or both. For an explanation of PLT processing, see “PLT Processing” on page 2-394.

- CICS tracing (optional)

DB2 uses the CICS trace and auxiliary trace facilities for tracing. See *Diagnosis Guide and Reference* for instructions on turning on tracing for the CICS Attachment. See the appropriate *CICS Problem Determination Guide* for instructions on using tracing.

- Paging

The attachment facility transaction DSNC uses standard BMS page retrieval, documented in the *CICS Application Programmer's Reference*. To change the default paging characters, use the PGRET option of the DFHSIT macro.

- DB2 program and transaction entries

If you are using the resource definition online (RDO), ensure that the GRPLIST option of the DFHSIT macro includes:

1. A group that contains entries for the CICS task-related user exit programs
2. A group that contains entries for the CICS attachment facility

For CICS Version 3 and earlier, job DSNTIJSU creates this group, and it has a default name of DB2. For CICS Version 4, the attachment facility definitions are in a CICS-defined group named DFHDB2.

If you are using macro-defined resources, enter the suffixes of the processing control table (PCT) and the processing program table (PPT), which contain the CICS attachment facility entries in the PCT= and PPT= SIT options.

- Storage Entries

See “Calculating Space Requirements for the CICS Attachment Facility” on page 2-381 for information about the CICS storage size system initialization parameters.

- Authorization Exits

If you are using AUTH=GROUP in the RCT and have a CICS release earlier than Version 4, you must specify EXTSEC=YES in the SIT and in the appropriate sign-on table (SNT) entries. For CICS Version 4, specify SEC=YES. For more information on group authorization, see “Coordinating DB2 and CICS Security” on page 2-396 and the GROUP entry on page 2-410.

- When using COBOL II
Specify COBOL2=YES.
- When using PL/I
Specify PLI=YES.
- Temporary Storage Table (TST)

The CICS attachment facility uses a temporary storage queue that has a request identifier of '.DSNTS'. To ensure that the data in this queue is not protected, the TYPE=SECURITY entries in the TST at your site should not specify a DATAID that uses any of the leading characters in '.DSNTS'. In other words, do not use DATAIDs that begin with '.', '.D', '.DS', '.DSN', '.DSNT', or '.DSNTS' in your TST TYPE=SECURITY entries.

Transaction Entries

To define your DB2 transaction entries to CICS, use either RDO or macro-defined PCT. For CICS Version 3, you must use RDO to define DB2 CICS transaction entries to CICS. The RDO group specified on installation panel DSNTIP3 is loaded with the RDO definitions, some examples of which are shown in Table 116 on page 2-388. As shown in the table, DSNCR is the only DB2-supplied transaction that is shipped with DB2. Refer to *Command Reference* for more details on the DSNCR subcommands, or the DB2 commands you can enter. For CICS Version 4, the attachment facility definitions are supplied automatically. You can use RDO to define sample applications and optional entries.

Resource Definition Online (RDO): If you are using resource definition online to define your transaction entries, see Table 116 on page 2-388 for an example of the transaction definitions you need. If you have specified at installation time that you are using CICS Version 3, the installation CLIST edits job DSNTIJSU to create these RDO entries in a separate CICS group for you. See “Using the Attachment Facility for CICS Version 3” on page 2-380 for more information.

- Tracing

Each transaction that you want CICS to trace must have TRACE=YES either specified or selected by default in the transaction entry. TRACE=YES is the CICS default.

- CICS resource manager interface definitions

The CICS-supplied resource group DFHRMI (included in the DFHLIST list) specifies the resource manager interface (RMI).

- DB2 commands

For CICS Version 3: You must have a transaction entry for transaction DSNCR, which executes program DSNCCOM1. DSNCCOM1 handles DB2 commands entered from a CICS terminal. In addition to a DSNCR entry, you can optionally define one or more DSNCR subcommands, or the first four characters of DB2 commands, as transactions that execute DSNCCOM1.

For CICS Version 4: Do not create a transaction entry for transaction DSNCL, which is automatically supplied by CICS. You can optionally define one or more DSNCL subcommands, or the first four characters of DB2 commands, as transactions that execute DSN2COM1.

For more information on DSNCL commands, see “Providing CICS Support for DB2 Commands” on page 2-395.

- DB2 sample applications

Phase 5 of installation verification requires that program, mapset, and transaction resources be defined to CICS. See Table 116 for a list of required entries. For more information on the verification procedure, see “Chapter 2-9. Verifying with the Sample Applications” on page 2-305.

- User transactions

User transactions accessing DB2 resources must also be defined to CICS.

Table 116 shows examples of transaction definitions.

Table 116 (Page 1 of 2). CICS Transaction Resource Definitions for DB2

Functions Defined to CICS	Resource Definition
DB2 commands entered as DSNCL subcommands	<pre>DEFINE TRANSACTION(DSNCL) PROGRAM(DSNCCOM1) GROUP(DB2-group) TWSIZE(1200)</pre> <p>See note 2 on page 2-389.</p>
DB2 Commands entered directly	<p>As many as wanted, for example:</p> <pre>DEFINE TRANSACTION(DISC) PROGRAM(DSNyCOM1) GROUP(DB2-group) TWSIZE(1200)</pre> <pre>DEFINE TRANSACTION(DISP) PROGRAM(DSNyCOM1) GROUP(DB2-group) TWSIZE(1200)</pre> <pre>DEFINE TRANSACTION(MODI) PROGRAM(DSNyCOM1) GROUP(DB2-group) TWSIZE(1200)</pre> <pre>DEFINE TRANSACTION(STOP) PROGRAM(DSNyCOM1) GROUP(DB2-group) TWSIZE(1200)</pre> <pre>DEFINE TRANSACTION(STRT) PROGRAM(DSNyCOM1) GROUP(DB2-group) TWSIZE(1200)</pre> <pre>DEFINE TRANSACTION(-DIS) PROGRAM(DSNyCOM1) GROUP(DB2-group) TWSIZE(1200)</pre> <p>(or any valid DB2-related command using a hyphen followed by the three character command abbreviation. Note that -STA DB2 cannot be issued from a CICS console. See <i>Command Reference</i> for all the valid commands. See also “Providing CICS Support for DB2 Commands” on page 2-395 for more information.)</p>
User transactions	<p>As many as needed:</p> <pre>DEFINE TRANSACTION(TRN1) PROGRAM(PROGRAM1) GROUP(user-group)</pre>

Table 116 (Page 2 of 2). CICS Transaction Resource Definitions for DB2

Functions Defined to CICS	Resource Definition
Sample application transactions during installation verification	<pre> DEFINE TRANSACTION(D8PS) PROGRAM(DSN8CP0) GROUP(DB2-group) DEFINE TRANSACTION(D8PP) PROGRAM(DSN8CP6) GROUP(DB2-group) DEFINE TRANSACTION(D8CS) PROGRAM(DSN8CC0) GROUP(DB2-group) DEFINE TRANSACTION(D8PT) PROGRAM(DSN8CP3) GROUP(DB2-group) DEFINE TRANSACTION(D8PU) PROGRAM(DSN8CP3) GROUP(DB2-group) </pre>

Note:

- To make the resource definitions available to CICS, issue the following commands:

```

CEDA INSTALL GROUP(DB2-group)
CEDA INSTALL GROUP(user)

```
- For CICS Version 4, do not create entries for DSNB commands. These are automatically defined by the new attachment facility.
- When using CICS Version 3 Release 2 Modification 1 or Version 3 Release 3, all user transactions accessing DB2 resources must be defined with the TASKDATALOC(BELOW) parameter. TASKDATALOC(ANY) is not supported for transactions accessing DB2 resources.
- For CICS Version 3 Release 2 Modification 1 or Version 3 Release 3, each transaction definition for PROGRAM(DSNCCOM1) should include the TASKDATALOC(BELOW) parameter, as in the following example:

```

DEFINE TRANSACTION(DSNC) PROGRAM(DSNCCOM1)
GROUP (DB2-GROUP) TWSIZE(1200)
TASKDATALOC(BELOW)

```

For CICS Version 4, transactions definitions for PROGRAM(DSN2COM1) should include the TASKDATALOC(ANY) parameter, as in the following example:

```

DEFINE TRANSACTION(DISC) PROGRAM(DSN2COM1)
GROUP (GROUP) TWSIZE(1200)
TASKDATALOC(ANY) TASKDATAKEY(CICS)

```
- These sample entries contain only the minimum options required. However, you might want to consider other options, such as using resource security levels (RSL) or adding the defined GROUPs into a GROUP LIST. For more information on options, see the appropriate CICS Resource Definition Guide.
- Ensure that you have the proper amount of TWA. If you do not, abend DFH0409 could result when you start up the attachment facility or enter a command through it.

CICS PCT Macro-defined Resources:

- Tracing Each transaction that you want CICS to trace must have TRACE=YES either specified or selected by default in the PCT. TRACE=YES is the default.
- CICS attachment facility entries

Set TYPE=GROUP, FN=RMI to generate entries required for CICS services that the CICS attachment facility needs. You must have a transaction entry for transaction DSNB that executes program DSNyCOM1. DSNyCOM1 handles DB2 commands entered from a CICS terminal.

For all CICS releases, you can optionally define one or more DSNB subcommands, or the first four characters of DB2 commands, as transactions which execute DSNyCOM1. For more information on DSNyCOM1, see "Providing CICS Support for DB2 Commands" on page 2-395.
- Dynamic transaction backout (DTB)

Specify DTB=YES for each transaction that accesses DB2 resources to activate dynamic transaction backout for those resources.

- DB2 CICS sample entries

DB2 sample entries are listed in *prefix.SDSNSAMP(DSN8FPCT)*.

- Transaction work area storage

Any transaction that is defined to execute program DSNyCOM1 must be defined with a TWASIZE of 1200.

For more information, see the appropriate CICS resource definition guide.

Program Entries

To define your DB2 program entries to CICS, use either RDO or macro-defined PCT. If you are running with CICS Version 3 or above, the RDO is the only means of defining DB2 CICS transaction entries to CICS. The RDO group specified on installation panel DSNTIP3 is loaded with the RDO definitions shown in Table 117 on page 2-391, and Table 118 on page 2-392.

Resource Definition Online (RDO): You can use RDO to create CICS definitions for CICS attachment facility modules, DB2 sample programs, and application programs. Table 117 on page 2-391 illustrates the program and map entries needed for CICS Version 3. Table 118 on page 2-392 illustrates the program and map entries needed for CICS Version 4. If you specified at installation time that you are using CICS 3.3, the installation CLIST edits job DSNTIJSU to create these RDO entries in a separate CICS group for you. For more information, see “Using the Attachment Facility for CICS Version 3” on page 2-380.

Ensure that you use the CICS-supplied resource group DFHRMI (included in the DFHLIST list) to specify the resource manager interface (RMI).

Table 117 (Page 1 of 2). CICS Version 3 Program and Map Resource Definitions for DB2

Functions Defined to CICS	Resource Definition
Attachment facility modules	<pre> DEFINE PROGRAM(DSNCCOM0) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCCOM1) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCCOM2) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCEDF1) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCEDON) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCEXT1) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCEXT2) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCMMSG0) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCSTOP) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCSTRT) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) DEFINE PROGRAM(DSNCUEXT) GROUP(DB2-group) LANGUAGE(ASSEMBLER) CEDF(NO) </pre>
User transactions	<pre> As many as needed: DEFINE PROGRAM(program-name) GROUP(user-group) LANGUAGE(prog-lang) </pre>
Sample application transactions during install verification	<pre> DEFINE PROGRAM(DSN8CP0) GROUP(DB2-group) LANGUAGE(PLI) DEFINE PROGRAM(DSN8CP1) GROUP(DB2-group) LANGUAGE(PLI) DEFINE PROGRAM(DSN8CP2) GROUP(DB2-group) LANGUAGE(PLI) DEFINE PROGRAM(DSN8CP3) GROUP(DB2-group) LANGUAGE(PLI) DEFINE PROGRAM(DSN8CP6) GROUP(DB2-group) LANGUAGE(PLI) DEFINE PROGRAM(DSN8CP7) GROUP(DB2-group) LANGUAGE(PLI) DEFINE PROGRAM(DSN8CP8) GROUP(DB2-group) LANGUAGE(PLI) DEFINE PROGRAM(DSN8CC0) GROUP(DB2-group) LANGUAGE(COBOL) DEFINE PROGRAM(DSN8CC1) GROUP(DB2-group) LANGUAGE(COBOL) DEFINE PROGRAM(DSN8CC2) GROUP(DB2-group) LANGUAGE(COBOL) </pre>

Table 117 (Page 2 of 2). CICS Version 3 Program and Map Resource Definitions for DB2

Functions Defined to CICS	Resource Definition
	DEFINE MAPSET(DSN8CPG) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPD) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPF) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPE) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CCG) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CCD) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPN) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPL) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPU) GROUP(DB2-group)

Note:

- To make the resource definitions available to CICS, issue the following install commands:

```
CEDA INSTALL GROUP(DB2-group)
CEDA INSTALL GROUP(user)
```
- These illustrated commands are the minimum options required. However, you might want to consider other options, such as resource security levels (RSL) or adding the defined GROUPs into a GROUP LIST. For more information on options, see the appropriate CICS resource definition guide.
- Attachment facility modules are required to have CEDF(NO); if they do not, the attach issues error messages DSNCO49I and DSNCO50I.
- If you use CICS 3.3, all definitions for programs related to the CICS attachment facility should include the DATALOC(BELOW) parameter, as in the following definition:

```
DEFINE PROGRAM(DSNCCOM1) GROUP(DB2-group)
LANGUAGE(ASSEMBLER)
CEDF(NO) DATALOCATION(BELOW)
```
- If you use CICS 3.3, all definitions for programs related to the CICS attachment facility should include the EXECKEY(CICS) parameter, as in the following definition:

```
DEFINE PROGRAM (DSNCCOM1) GROUP(DB2-group)
LANGUAGE(ASSEMBLER)
CEDF(NO) EXECKEY(CICS)
DATALOCATION(BELOW)
```
- The sample user exit for dynamic plan selection, DSNCUEXT, is not always required.

Table 118 (Page 1 of 2). CICS Version 4 Program and Map Resource Definitions for DB2

Functions Defined to CICS	Resource Definition
User transactions	As many as needed: <pre>DEFINE PROGRAM(program-name) GROUP(user-group) LANGUAGE(prog-lang)</pre>

Table 118 (Page 2 of 2). CICS Version 4 Program and Map Resource Definitions for DB2

Functions Defined to CICS	Resource Definition
Sample application transactions during install verification	DEFINE PROGRAM(DSN8CP0) GROUP(DB2-group) LANGUAGE(PLI)
	DEFINE PROGRAM(DSN8CP1) GROUP(DB2-group) LANGUAGE(PLI)
	DEFINE PROGRAM(DSN8CP2) GROUP(DB2-group) LANGUAGE(PLI)
	DEFINE PROGRAM(DSN8CP3) GROUP(DB2-group) LANGUAGE(PLI)
	DEFINE PROGRAM(DSN8CP6) GROUP(DB2-group) LANGUAGE(PLI)
	DEFINE PROGRAM(DSN8CP7) GROUP(DB2-group) LANGUAGE(PLI)
	DEFINE PROGRAM(DSN8CP8) GROUP(DB2-group) LANGUAGE(PLI)
	DEFINE PROGRAM(DSN8CC0) GROUP(DB2-group) LANGUAGE(COBOL)
	DEFINE PROGRAM(DSN8CC1) GROUP(DB2-group) LANGUAGE(COBOL)
	DEFINE PROGRAM(DSN8CC2) GROUP(DB2-group) LANGUAGE(COBOL)
	DEFINE MAPSET(DSN8CPG) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPD) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPF) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPE) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CCG) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CCD) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPN) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPL) GROUP(DB2-group)
	DEFINE MAPSET(DSN8CPU) GROUP(DB2-group)

Note:

- To make the resource definitions available to CICS, issue the following install commands:

```
CEDA INSTALL GROUP(DB2-group)
CEDA INSTALL GROUP(user-group)
```
- These illustrated commands are the minimum options required. However, you might want to consider other options, such as resource security levels (RSL) or adding the defined GROUPs into a GROUP LIST. For more information on options, see the appropriate CICS resource definition guide.

Table 119 on page 2-394 shows examples of the optional PLT entries that allow automatic initialization of the CICS attachment facility when CICS is brought up and automatic stopping of the CICS attachment facility when CICS is shut down.

CICS PPT Macro-Defined Resource:

- Specify TYPE=GROUP,FM=RMI to generate entries required for CICS services that the CICS attachment facility needs. In addition, specify DFHPPT TYPE=GROUP,FN=INQUIRESET.
- For CICS releases earlier than Version 4, create an entry for program DSNCCOM1 to support DB2 command processing from a CICS terminal.

- If you are using VS COBOL II, create entries for the VS COBOL II library routine definitions. Refer to *VS COBOL II Installation and Customization for MVS* for more information.
- DB2 sample entries are listed in *prefix.SDSNSAMP(DSN8FPPT)*.
- When using a program list table (PLT) to automatically start or stop the attachment facility, you must define the PLT in the CICS PPT. For more information on PLT processing, see “PLT Processing.”

PLT Processing

The program list table (PLT) can be used to specify a list of programs that are:

- Executed in the post-initialization phase of CICS startup
- Executed during controlled shutdown
- Enabled or disabled as a group by a master terminal ENABLE or DISABLE command.

Optionally, you can use PLTs to have CICS automatically bring up the DB2-CICS attachment facility during the post-initialization phase of CICS startup. Similarly, you can have CICS terminate the attachment facility when CICS is brought down.

To use PLT processing to bring up the attach during CICS initialization:

- Create a PLTPI with an entry for PROGRAM=DSNyCOM0.
- Create a PLTPI=xx entry in the SIT, where xx is the suffix of the post-initialization PLT.

To use PLT processing to automatically shut down the attach during CICS shutdown:

- Create a PLTSD with an entry for PROGRAM=DSNyCOM2.
- Create a PLTSD=xx entry in the SIT, where xx is the suffix of the termination PLT. Use separate PLTs for post-initialization and termination because separate DB2 modules are used for starting and stopping the attachment facility.

Refer to “Transaction Entries” on page 2-387 and “Program Entries” on page 2-390 for more information on required entries for PLT processing.

For a list of optional entries on the PLT, refer to Table 119.

Table 119 (Page 1 of 2). Optional CICS Program List Table (PLT) Entries for DB2

Functions Defined to CICS	PLT Entry
Post-initialization attachment program	DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM : DFHPLT TYPE=ENTRY,PROGRAM=DSNyCOM0
First quiesce stage shutdown program	DFHPLT TYPE=ENTRY,PROGRAM=DSNyCOM2 : DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM

Table 119 (Page 2 of 2). Optional CICS Program List Table (PLT) Entries for DB2

Functions Defined to CICS	PLT Entry
---------------------------	-----------

Note:

1. DSNyCOM0 must be defined in the last stage of CICS initialization (PLTPI).
2. DSNyCOM2 must be defined in the first quiesce stage of CICS shutdown (PLTSD). To ensure that DSNyCOM2 executes during the first quiesce stage, be sure that the DSNyCOM2 program entry is defined before the DFHDELIM program entry.

For CICS Version 3.3 and earlier, to access DB2 tables at CICS start-up time, you cannot issue SQL statements directly from your PLT program. Your PLT program must start another transaction that issues the SQL statements. The following example shows one method of starting another transaction with an interval. The interval specified should be long enough to insure that the attachment facility has had enough time to complete its start-up.

```
EXEC CICS START TRANSID('xxxx') INTERVAL(hhmmss)
```

For CICS Version 4, you can issue SQL statements from your PLT program. Make sure that the SQL statements are after DSN2COM0 for startup and before DSN2COM2 for shutdown. You cannot start the attachment facility during CICS shutdown.

Providing CICS Support for DB2 Commands

DB2-related commands can be entered from a CICS terminal. These can be commands to the attachment facility or to the DB2 subsystem. The commands are handled by a DB2-supplied program, DSNyCOM1.

DSNyCOM1 scans the first two input fields for the command verb. Therefore, you can choose to place a CICS transaction ID in front of the command verb, or you can make the command verb the transaction ID. For example, the DISP entry in Table 116 on page 2-388 allows a user to enter DISP as a transaction ID. DSNyCOM1 recognizes this request as equivalent to DSNyCOM1 DISP. In addition, it is also possible to create transaction entries for DSNyCOM1 that consist of the first four characters of valid DB2 commands, such as -DIS or -REC, so a user can enter a DB2 command such as -DSN1 DISPLAY THREAD(*) from a CICS terminal. DB2 commands and DSNyCOM1 subcommands are documented in *Command Reference*.

To provide support for command processing, you must:

- Define a program resource to CICS using RDO for the command processor, DSNyCOM1.
- Ensure that you have a definition for the command processing transaction, DSNyCOM1.
- Define any other transaction resources to CICS for DB2 commands you want to use directly as transaction IDs. You must also correlate these entries to the DSNyCOM1 command processor program.
- Authorize the CICS user ID and the sign-on operator ID, the terminal ID to issue DB2 commands, or both.

The CICS attachment facility uses the same authorization scheme for commands that it uses for all other transactions; for this reason, authorization must be granted according to your chosen authorization scheme. It is possible to change the command's authorization directives by specifying an entry in the RCT (DSNCRCT

TYPE=COMD). For information about levels of authority needed to issue DB2 commands, see the description of each command in *Command Reference* .

Updating CICS Initialization JCL

DB2 libraries: If your DB2 program library (SDSNLOAD) is different from your RCT library, you need to specify both libraries in the STEPLIB/JOBLIB concatenation in your CICS startup job or in the LINKLSTxx.

For CICS Version 3 Release 3 and Earlier: The DB2 program library must be included in the CICS startup DFHRPL concatenation. If the DB2 program library is not in the LINKLSTxx, the libraries containing your RCT, or modules DSNyEXT3, DSNyMSUB, DSNCSM21, and DSNCSM31 must be concatenated with the STEPLIB or JOBLIB libraries because they are loaded using MVS services, not CICS services. When you concatenate these libraries, remember that STEPLIB and JOBLIB libraries must be APF-authorized to maintain the APF authorization for the other libraries in STEPLIB.

For CICS Version 4: DB2 does not need the DB2 program libraries in the DFHRPL DD statement. If you do include DB2 program libraries in the STEPLIB or DFHRPL DD concatenations for other applications, you must place them after the CICS program libraries. You can no longer use DSNCRCTx on the DFHSIP parameter to specify the RCT suffix. The DSNCRCTx initialization option has been replaced the INITPARM= option.

Region Size: Increase the region size by the amount calculated for the local system queue area (LSQA). For the LSQA calculation, refer to “Calculating Space Requirements for the CICS Attachment Facility” on page 2-381.

RCT Suffix: The attachment facility allows multiple RCTs to coexist by using a suffix field for identification. The default RCT suffix is 0 for Version 3.3 and earlier and 00 for Version 4. For more information on using the RCT suffix, see page 2-404.

Coordinating DB2 and CICS Security

The authorization ID sent to DB2 from CICS can be any one of the following:

- The CICS sign-on user ID (USERID)

For CICS Version 3 Release 3 and Earlier: CICS can use the USERID if the transaction is invoked from a signed-on CICS terminal. This value is defined in the CICS sign-on table. This authorization offers the advantage of being verified by RACF if the correct CICS options are chosen. For more information, see the appropriate CICS resource definition guide.

For CICS Version 4: CICS can use a default USERID if you are not signed on. CICS Version 4 does not use a sign-on table.

- The CICS sign-on operator ID (OPIDENT)

For CICS Version 3 Release 3 and Earlier: CICS can use OPIDENT if the transaction is invoked from a CICS terminal at which the user has signed on. This value is defined in the CICS sign-on table.

For CICS Version 4:

Specify the OPIDENT using the CICS option of the RACF ADDUSER command. For more information, see *Resource Access Control Facility (RACF) Command Language Reference*.

- The CICS terminal name (TERM)
- The CICS transaction ID (TXID)
- The user-defined ID (SIGNID) in macro DSNCRCCT TYPE=INIT.
- The group authorization ID (GROUP) in macro DSNCRCCT TYPE=ENTRY if the RACF user ID and connected group name are passed to DB2 for security checking.
- Any user-defined ID up to eight characters long, except for the reserved keywords USERID, USER, TERM, TXID, SIGNID, and GROUP.

The value used depends on the AUTH option specified in macro DSNCRCCT TYPE=COMD, TYPE=ENTRY and TYPE=POOL. This option supports up to three directives for selecting an authorization ID.

The attachment facility scans the supplied list until it finds a value to be used as the authorization ID or it until it reaches the end of the list. With CICS Version 3, for example, an AUTH=(USER,TERM,TXID) specification for a terminal-driven transaction with no sign-on causes the terminal ID to be used as an authorization ID. For a non-terminal-driven transaction, the transaction ID is used. AUTH=USERID behaves differently in CICS Version 4 than it did in prior releases for non-terminal-driven transactions. CICS Version 4 has a default user ID, so AUTH=USERID is a valid specification, even for transactions without a terminal. If RACF is installed, you also need to define the CICS-to-DB2 connection to RACF. For information on how to do this, see Section 3 (Volume 1) of *Administration Guide*.

The ability to enter DB2 and attachment commands from CICS terminals can also be controlled through standard CICS security mechanisms. See Section 4 (Volume 1) of *Administration Guide* for information on the DB2 commands that can be issued in the CICS environment.

Preparing CICS Applications for DB2

The following tasks must be performed for CICS applications accessing DB2 resources:

1. Precompile the application using the DB2 precompiler.
2. Translate the application using the CICS command translator.
3. Compile or assemble the application.
4. Link-edit the application, including the DB2-supplied SQL language interface (DSNCLI) and the appropriate CICS command level interface module. If the application also accesses DL/I, the corresponding IMS language interface must also be included.
5. Bind the application to its application plan.

If you want to determine whether the attach facility is available, you can use the EXTRACT EXIT command.

EXTRACT EXIT determines whether the global work area exists for program 'DSNCEXT1' with the entry name 'DSNCSQL'. If the command response is an invalid exit request (INVEXITREQ), then the attachment facility is not available. However, a command response of NORMAL does not *guarantee* that the attachment facility is available because the exit might not have been ENABLE STARTED. CICS Version 4 has another command which allows you to determine whether the attachment facility has been enable-started:

```
EXEC CICS INQUIRE EXITPROGRAM
```

The INQUIRE EXITPROGRAM CONNECTST parameter should be used to test whether the attachment facility is ready to process SQL statements.

If your applications call DSNTIAC, make sure you have made the changes mentioned in “Using CICS Storage-Handling Facilities” on page 2-331.

For more information about preparing CICS applications for DB2, see Section 4 of *Application Programming and SQL Guide*.

CICS Attachment Facility Macro (DSNCRCT)

The macro description in this chapter has a set of diagrams that shows the syntax of the macro and guides you through its coding.

The DSNCRCT macro is used to specify the CICS resource control table (RCT).

For CICS Version 3 Release 3 and earlier, use the DSNCRCT macro in your DB2 macro library, *prefix.SDSNMACS*. For CICS Version 4, use the DSNCRCT macro in the CICS410.SDFHMAC library.

This macro supports DSNTJ5A, which assembles and links the RCT. See “Job DSNTJ5A” on page 2-327 for more information. If you use the CICS-supplied procedures, you must make the following changes:

- Add the macro library (*prefix.SDSNMACS* or CICS410.SDFHMAC) to the SYSLIB DD statement in the assembly step.
- Override the SYSLMOD statement of the link-edit step to point to an APF-authorized library that is available to CICS at execution time.
- Override the program name for the SMP/E step by specifying the following option when you invoke the procedure:

```
SMPPGM=IEFBR14
```

The RCT is coded in the format shown below and then assembled.

If you are migrating and you use the CICS attachment facility, you must assemble and link-edit the RCT. This is optional, however, if you are using CICS version 4 or above.

The TYPE=INIT macro must be specified first. If the TYPE=COMD macro is specified, it must be second. If TYPE=POOL is specified, it must be after TYPE=INIT and TYPE=COMD (if there is a COMD) and before TYPE=ENTRY. The TYPE=ENTRY macro can be specified an indefinite number of times to identify specific transactions. The TYPE=FINAL macro is specified last, causing the RCT to be generated. For example:

```

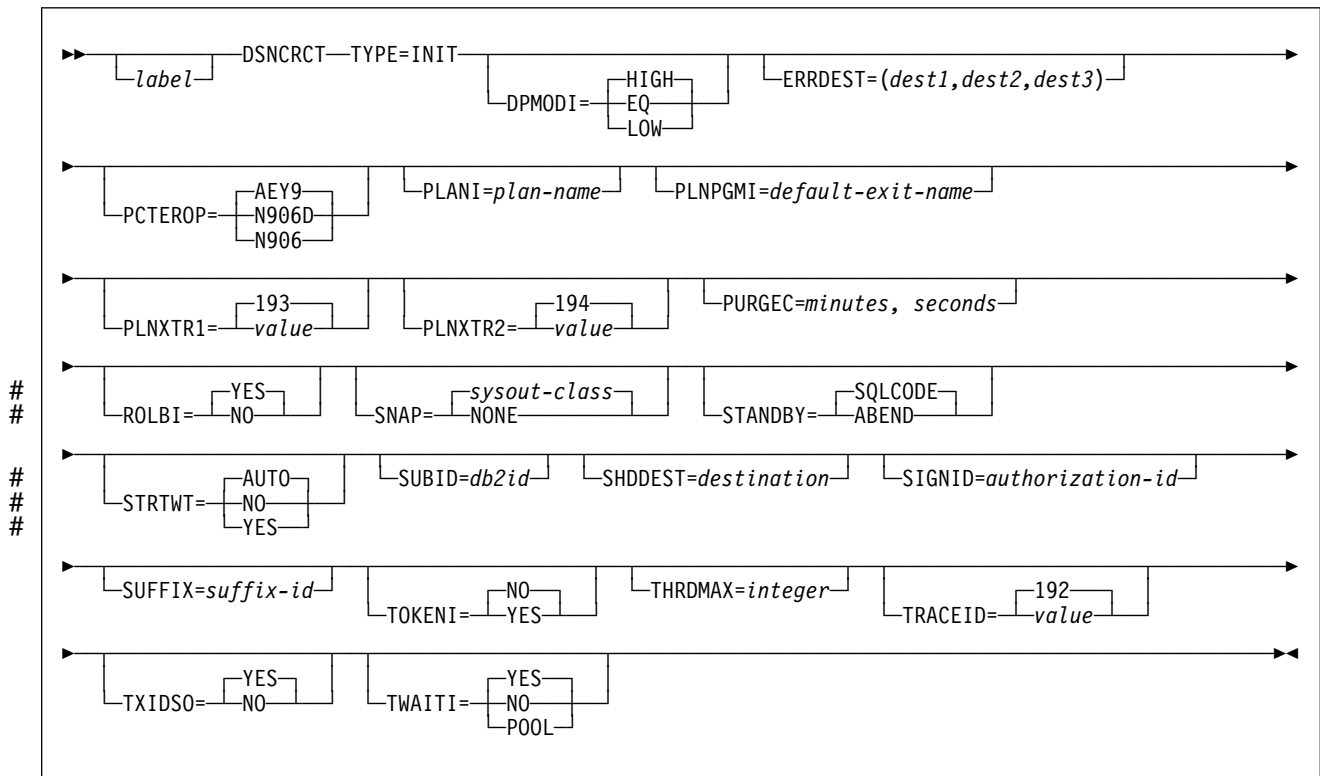
DSNCRCT TYPE=INIT,
DSNCRCT TYPE=COMD,[optional]
DSNCRCT TYPE=POOL,[optional]
DSNCRCT TYPE=ENTRY,[optional]
:
DSNCRCT TYPE=ENTRY,[optional]
DSNCRCT TYPE=FINAL
END      [assembler end statement required]

```

The RCT must be link-edited into a library that is accessible to MVS through its normal library search order (STEPLIB, JOBLIB, link library), and it must be link-edited below the 16MB line. Do not link the RCT as REENTRANT or REUSABLE because this causes the CICS attachment facility to abend on startup.

DSNCRCT TYPE=INIT

The TYPE=INIT macro allows you to define the information required for CICS to establish its first connection to DB2 and to specify the default options for other types of the macro.



Notes:

- The STANDBY keyword and the AUTO option of STRTWT are valid for CICS Transaction Server 1.1 and above. STRTWT=YES is the default for CICS Version 4 and earlier.
- The macro name must be followed by one or more blanks before options that are optional are coded.
- Multiple options must be separated by commas (with no blanks).

- Code a non-blank character in column 72 to indicate that the current statement is continued on the next line.
- RCT macro continuations should begin in column 16.

Description

DPMODI=HIGH|EQ|LOW

Specifies a default for the DPMODE parameter in other TYPES of this macro. The specification (HIGH, EQ, LOW) indicates the priority of thread subtasks relative to CICS main task priority.

HIGH

Specifies that subtasks can attain a higher priority than the CICS main task from which the subtask was generated. Use this option for high priority and high volume transactions.

EQ

Specifies that CICS must allow for subtasks to attain equal priority.

LOW

Specifies that subtasks have a lower priority than the CICS main task priority.

Default: HIGH

ERRDEST=(dest1,dest2,dest3)

Specifies up to three CICS transient data destinations to receive unsolicited messages. For *dest1,dest2,dest3*, substitute up to three valid transient data destinations.

An asterisk can be specified as a destination. The asterisk acts as a place holder and allows later specification of a destination by the DSNCS MODIFY DESTINATION command.

Destinations specified by this parameter are verified as valid transient data destinations when the attachment facility is started. Any destinations that are found to be invalid are changed to * (or CSMT, if none are valid).

Default: (CSMT,*,*)

PCTEROP=AEY9|N906D|N906

Specifies the type of processing that is to occur following a create thread error. The error processing occurs after the SQLCA's SQLCODE field has been updated to reflect the reason for the create thread failure. The PCTEROP parameter allows a user to specify whether a DSNCS transaction dump is taken, and whether the DSNCS SQL RMI associated with the transaction is disabled. This parameter can be used to allow a transaction to continue processing if a create thread error occurs. A transaction that continues after a create thread error must take corrective action to allow a new thread to be created. A SYNCPOINT ROLLBACK command must be part of the corrective action taken by the transaction before it can continue to issue SQL requests.

AEY9

Specifies that a DSNCS transaction dump is to be taken and the DSNCS SQL RMI associated with the transaction is to be disabled. Disabling the DSNCS SQL RMI causes the transaction to receive an AEY9 abend if it attempts to issue another SQL request. The transaction must be terminated and reinitialized before it is allowed to issue another SQL request.

N906D

Specifies that a DSNC transaction dump is to be taken and the DSNC SQL RMI associated with the transaction is **not** to be disabled. The transaction receives a -906 SQLCODE if another SQL is issued, unless the transaction issues SYNCPOINT ROLLBACK. SYNCPOINT without the ROLLBACK option results in an ASP7 abend.

N906

Specifies that a DSNC transaction dump is **not** to be taken and the DSNC SQL RMI associated with the transaction is **not** to be disabled. The transaction receives a -906 SQLCODE if another SQL request is issued, unless the transaction issues a SYNCPOINT ROLLBACK. SYNCPOINT without the ROLLBACK option results in an ASP7 abend.

Default: AEY9

PLANI=*plan-name*

Specifies the default plan name for any entry in the RCT that does not use dynamic plan selection. The *plan-name* can have 1-8 characters. Without the PLANI option, the plan name for an entry in the RCT is:

The value for PLAN= in the TYPE=ENTRY macro

The value for TXID= in the TYPE=ENTRY macro if PLAN= is not specified

With the PLANI option, the plan name for an entry in the RCT is:

The value for PLAN= in the TYPE=ENTRY macro

The value for PLANI= in the TYPE=INIT macro if PLAN= is not specified

The value for TXID= in the TYPE=ENTRY macro if neither PLAN= nor PLANI= is specified

The PLANI option has no effect on entries that use dynamic plan selection.

This parameter is only valid for CICS 4 and subsequent releases.

#

PLNPGMI=*default-exit-name*

Specifies the name of the default dynamic plan exit. If one of the entries has PLNEXIT=YES, but does not supply a value for PLNPGME, this parameter is used as the exit program name for that entry.

Default: DSNCUEXT

PLNXTR1=*integer*

Specifies the dynamic plan entry trace ID. Before passing the parameter list to the exit program, the attachment facility generates a CICS trace entry so that the plan name can be recorded in the CICS trace table. For *integer*, substitute a number from 0 to 199.

Default: 193

PLNXTR2=*integer*

Specifies the dynamic plan exit trace ID. After returning from the exit program, the attachment facility generates another CICS trace entry. This value is the trace ID assigned to that trace. For *integer*, substitute a number from 0 to 199.

Default: 194

PURGEC=*minutes,seconds*

Specifies the length of the protected thread purge cycle. The maximum value for PURGEC is (59,59). The minimum is (0,30).

An unprotected thread is terminated as soon as the transaction ends (at SYNCPOINT or EOT). A protected thread is terminated after two purge cycles, which are 30 seconds by default. Normally, a protected thread remains connected for 30-60 seconds after the transaction ends.

You can use PURGEC to modify the 'normal purge cycle'. The purge cycle is 5 minutes long when the attachment starts and then PURGEC for the remaining time that the attachment facility operates. For example, if you specify PURGEC=(0,40), protected threads are normally purged 40-80 seconds after the transaction ends.

#

This parameter is only valid for CICS 4 and subsequent releases.

Default: (0,30)

ROLBI=YES|NO

Specifies a default for the ROLBE parameter in other TYPEs of the DSNCRCCT macro. The specification of this parameter determines the disposition of transaction entries in the event a transaction is selected by DB2 as victim in a deadlock resolution.

YES

Specifies that the attachment facility is to issue a sync point rollback before returning control to the application. A SQL return code of -911 is returned to the program. Specifying YES provides compatibility with SQL/DS.

NO

Specifies that the attachment facility is not to initiate a rollback for this transaction. A SQL return code of -913 is returned to the application. It is the responsibility of the application to initiate the rollback.

Default: YES

SHDDEST=*destination*

Specifies a transient data destination to receive the statistical report (the same report that is displayed with the DSNC DISP STAT command) during shutdown of the attachment facility. For *destination*, substitute a valid transient data destination.

Default: CSSL

It might be useful to direct this transient output data to a destination in another partition that is specified as a JES SYSOUT file.

SIGNID=*authorization ID*

Specifies the authorization ID to be used by the CICS attachment facility when signing on to DB2. For *authorization ID*, substitute a character string of up to eight characters.

The default is the application name of the CICS system, as specified during CICS table generation using the APPLID operand. This name is used when indicated by the AUTH parameter of the TYPE=ENTRY or TYPE=POOL forms of the macro. See the description of the AUTH parameter under "DSNCRCCT TYPE=ENTRY" on page 2-409. When it is used, the name specified here must be authorized to the resources being accessed.

SNAP=sysout-class|NONE

Specifies the SYSOUT class that the attachment facility must use for taking a snap dump if a thread subtask fails. For *sysout-class*, substitute any class valid to MVS.

NONE

Specifies that the snap dump is not taken. However, the SDWA (fixed, base, variable areas, and extensions) is written to SYS1.LOGREC.

Default: A

#

STANDBY=SQLCODE|ABEND

Specifies the action to be taken by the attachment facility during the startup process if DB2 is not active. This keyword is valid for CICS Transaction Server 1.1 and above .

#

#

ABEND

Specifies CICS applications using DB2 fail with abend AEY9 issued by CICS when the attachment is not started.

SQLCODE

Only valid if STRTWT=AUTO is also specified. If an application issues a SQL statement while the attachment facility is standing by, SQLCODE -923 is issued instead of abend AEY9.

#

Default: SQLCODE

#

STRTWT=AUTO|NO|YES

Specifies the action to be taken by the attachment facility during the startup process if DB2 is not active.

YES

Directs the attachment facility to wait for DB2 to start and complete the connection. A CICS task waits to be posted by DB2 when DB2 becomes available. At that time, the initialization of the CICS attach completes. However, the attachment facility can be terminated by the DSNB STOP command while it is waiting for DB2.

The response messages from the attachment are sent to the transient data destination queue specified in the ERRDEST parameter of the RCT.

NO

Directs the attachment facility to terminate the connection process immediately if DB2 is not already active.

AUTO

Specifies to automatically restart the attachment facility if DB2 stops or abends, then restarts. The starting procedures are the same as for YES. If DB2 stops or abends while the attachment facility is up, a message is issued stating the subsystem is not active. The attachment facility goes to standby state and only terminates after the command DSNB STOP is issued or a fatal error is encountered. This keyword is valid for CICS Transaction Server 1.1 and above.

#

#

#

Default: AUTO

SUBID=DB2-ID

Specifies the name of the DB2 subsystem that the attachment facility is to connect with CICS. For *DB2 ID*, substitute a character string of up to four characters.

Default: DSN

SUFFIX=suffix-ID

Specifies the suffix identification character that is to be added to DSNCRCT when the CSECT name is generated. The suffix is one character for CICS Version 3 and earlier, and two characters for CICS Version 4. If it is loaded as a result of this suffix being entered as a parameter using the DSNCRCT command, the table must also be link-edited by this name. For more information on this command, see *Command Reference*.

For CICS Version 3 and earlier, you may define a suffix for the RCT when starting the attachment facility using PLT processing. To do so, specify the character string DSNCRCTx as the first parameter on the PARM field of the CICS EXEC DFHSIP JCL, where x represents the suffix of the desired RCT. For example:

```
//CICS EXEC PGM=DFHSIP,PARM=
('DSNCRCT2,SIT=21,SYSDIN')
```

Restrictions concerning this method of suffix specification are:

- The DSNCRCTx parameter must be in the PARM field of the EXEC JCL statement. The DSNCRCT x parameter is not supported as part of the SYSDIN data stream.
- The parameter must follow applicable MVS JCL rules with regard to parameters on the EXEC statement.
- It must adhere to CICS rules, such as the restriction that SYSDIN(SI) and CONSOLE(SN) must be the final parameter in the list.

For CICS Version 4, you can use the INITPARM command to select the RCF suffix. For example:

```
INITPARM=(DSN2STRT='xx')
```

You can also use the DSNCRCT command. For example:

```
DSNCRCT xx
```

For more information on CICS suffixes, see the appropriate CICS resource definition guide.

Default: 0 (CICS Version 3 and earlier), 00 (CICS Version 4)

THRDMAX=integer

Specifies the maximum number of subtasks that can be active before the attachment facility begins terminating inactive subtasks. The maximum number controls the use of main storage by MVS subtasks created for each active thread. When the number of created subtasks reaches THRDMAX-2, and there is a demand for more subtasks, the attachment facility terminates all the subtasks that are currently inactive. For that reason, the recommended value for THRDMAX is the sum of all values on the THRDA parameters (COMD, ENTRY, and POOL threads) + 3. However, the value you specify for THRDMAX can be less than the sum of all values on the THRDA parameter.

The value specified by this parameter must be at least the sum of all values on the THRDS parameters (COMD, ENTRY, and POOL threads). The minimum value is 4. The recommended maximum value is 2000. The value you choose for THRDMAX can be exceeded by increasing THRDA values for selected subtasks.

When determining the amount for THRDMAX, be sure to consider the amount you specified for the MAX USERS parameter on installation panel DSNTIPE. MAX USERS is described on page 2-145.

Default: 12

TOKENI=NO|YES

Specifies the default TOKENE if TOKENE is not specified on the TYPE=ENTRY statement. For more information on TOKENE, see page 2-414.

TRACEID=*trace-ID*

Specifies the CICS user trace ID that the CICS attachment facility is to use for tracing calls to DB2. Any value, from 0 through 199, that is valid to CICS and that does not conflict with values already in use for CICS trace can be specified.

Default: 192

TXIDSO=YES|NO

Specifies whether you want to suppress some sign-ons during thread reuse, and thereby avoid extraneous accounting information. The TXIDSO option affects only pool threads and those RCT entry threads with multiple transaction IDs in one entry (for example, TXID=(XC05,XC07). The attach checks for thread reuse only within an entry. TXIDSO has no effect on transactions that specify TOKENE=YES. If the plan name changes, the thread is terminated and re-created. This parameter is only valid for CICS 4 and subsequent releases.

YES

Specifies the following rules for thread reuse:

1. A new transaction can reuse an existing thread without a sign-on when:
The authorization ID and transaction ID are the same as the last transaction that used the thread, and TOKENE is set to NO.
2. A new transaction must sign-on before reusing an existing thread when any of these conditions exist:
The authorization ID is different from the authorization ID that last used the thread.
TOKENE is YES.
The transaction ID has changed.

NO

Specifies the following rules for thread reuse:

1. A new transaction can reuse an existing thread without a sign-on when:
The authorization ID is the same as the last transaction that used the thread and TOKENE is set to NO.
2. A new transaction must sign-on before reusing an existing thread when either of these conditions exist:
The authorization ID is different from the authorization ID that last used the thread.
TOKENE is YES.

Default: YES

TWAIT=YES|NO|POOL

Specifies the default value (YES, NO, or POOL) that is to be created for the TWAIT parameter on other types of the macro.

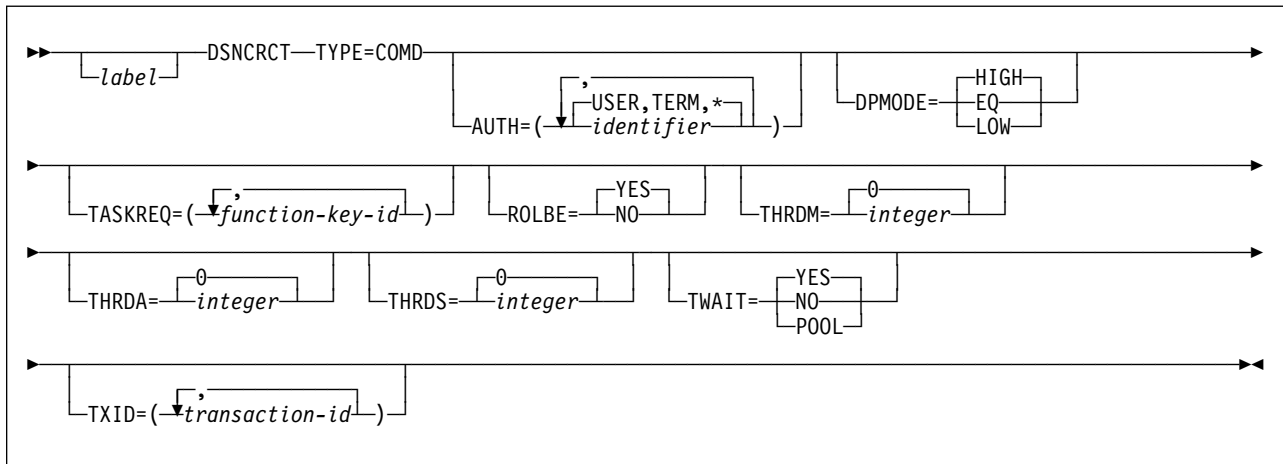
Default: YES

For a description of the TWAIT parameter, refer to “DSNCRCT TYPE=ENTRY” on page 2-409.

DSNCRCT TYPE=COMD

The TYPE=COMD form of the macro is used to describe DB2 command threads used for processing DB2 commands only. Requests for a DB2 command thread can be diverted to the pool (TWAIT=POOL) if a DB2 command thread is not available.

If you code the DSNCRCT TYPE=COMD macro, you must place it immediately following the TYPE=INIT macro and before the TYPE=POOL macro.



Notes:

- The macro name must be followed by one or more blanks before optional parameters are coded.
- Multiple parameters must be separated by commas (with no blanks).
- Code a non-blank character in column 72 to indicate that the current statement is continued on the next line.
- RCT macro continuations should begin in column 16.

Description

The TYPE=COMD form of DSNCRCT is optional; if you do not code it, the RCT automatically generates a TYPE=COMD entry with the following parameters:

```
DSNCRCT TYPE=COMD, DPMODE=HIGH, TXID=DSNC,  
  AUTH=(USER, TERM, *),  
  ROLBE=NO,  
  THRDM=1,  
  THRDA=1,  
  THRDS=1,  
  TWAIT=POOL
```

One reason you might want to change this generated default setting is to change the AUTH parameter to reflect the need to give users different responsibilities in operating the DB2 subsystem. The AUTH parameter is described under TYPE=ENTRY on page 2-409.

We strongly recommend the generated values THRDM=1, THRDA=1, THRDS=1, and TWAIT=POOL, because command threads are normally very low use threads. Defining more than one command thread probably unnecessarily wastes resources. In the event a second command thread is needed, it can be diverted to the pool. We also recommend ROLBE=NO for command threads, because rollback is not used for DB2 commands.

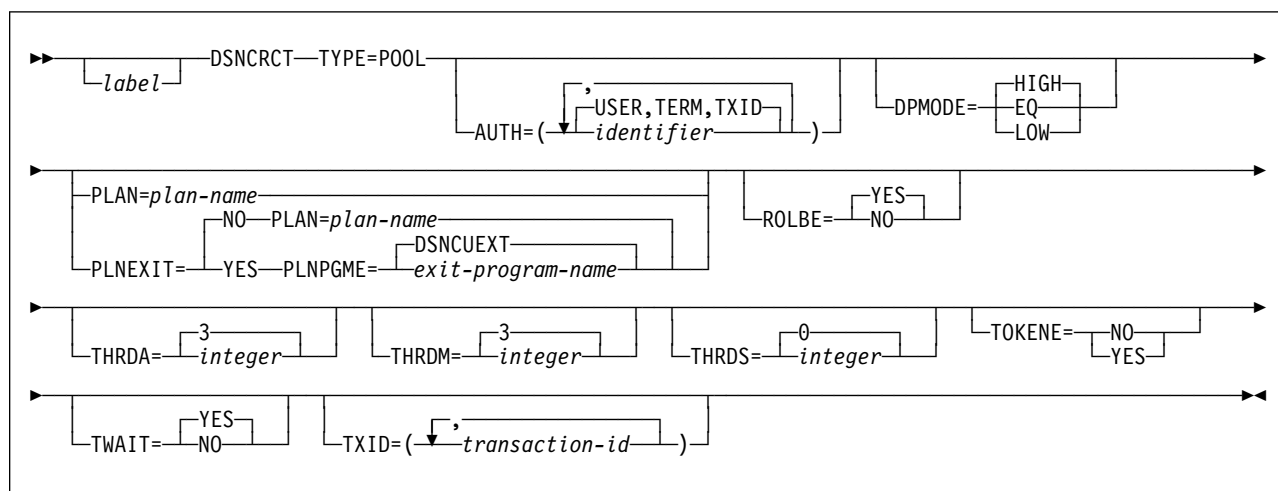
It is necessary only to specify DSNCRCT (or your user-defined command thread transaction) for the TXID parameter, even if you are using multiple transaction IDs to execute DSNCCOM1.

For more information on the TYPE=COMD parameters, see the descriptions under TYPE=ENTRY on page 2-409.

DSNCRCT TYPE=POOL

The TYPE=POOL form of the macro defines threads that can be shared by some or all of the CICS transactions. These threads are allocated to transactions only for a CICS unit of work. They can be considered short-term threads, and they can be used by any RCT entry that is specified to overflow to the pool.

The TYPE=POOL form of the macro must be coded before the TYPE=ENTRY form.



Notes:

- The macro name must be followed by one or more blanks before optional parameters are coded.
- Multiple parameters must be separated by commas (with no blanks).
- Code a non-blank character in column 72 to indicate that the current statement is continued on the next line.
- RCT macro continuations should begin in column 16.

Description

DSNCRCT TYPE=POOL

This is the name of the macro. It must be coded exactly as it appears here.

If the TYPE=POOL form of the macro is not coded, the following default values are assumed:

```
DSNCRCT TYPE=POOL,PLAN=DEFAULT,  
        AUTH=(USER,TERM,TXID),  
        TXID=POOL,  
        THRDM=3,  
        THRDA=3,  
        THRDS=3,  
        TWAIT=YES
```

This default assumes there is a plan named DEFAULT.

The rest of the parameters on the TYPE=POOL form of the macro are basically the same as those on the TYPE=ENTRY form of this macro. The only differences for the parameter specifications on TYPE=POOL are:

- The default for PLAN is DEFAULT.
- The default for both THRDM and THRDA is 3 (instead of 0 for TYPE=ENTRY).
- The TWAIT parameter does not include TWAIT=POOL.
- The default for TXID is POOL.
- The THRDS parameter specifies the number of MVS subtasks to start when the attachment facility is started, but no pool threads are protected.

For a description of the parameters, see “DSNCRCT TYPE=ENTRY” on page 2-409.

Usage Notes:

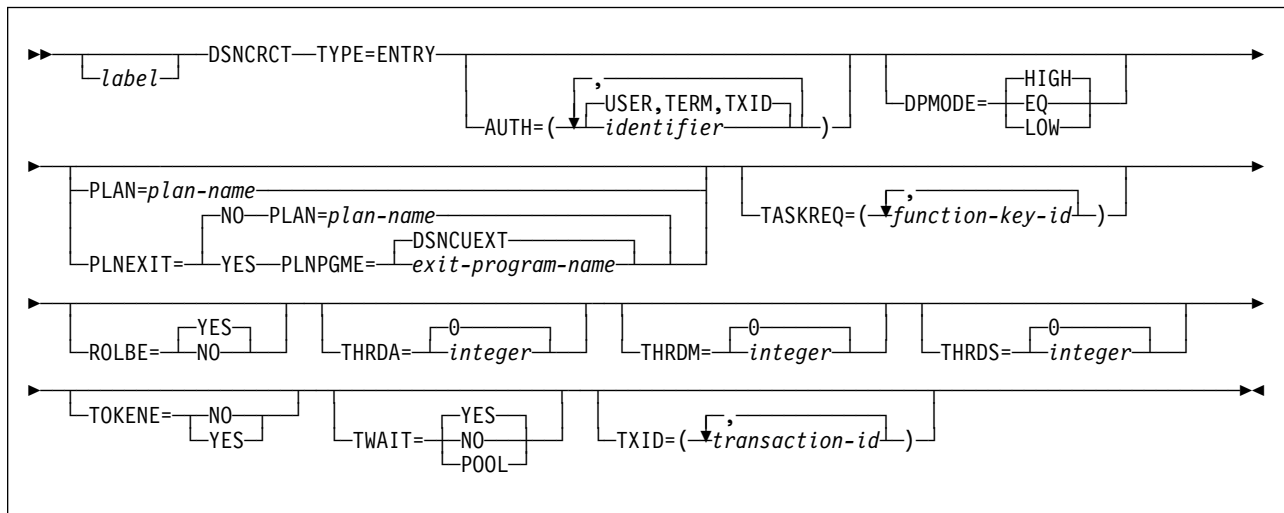
- When the pool is used to create a thread for a transaction, the thread is terminated when the transaction reaches SYNCPOINT (terminal-driven transactions) or EOT (non-terminal-driven transactions), unless another transaction is queued for the thread with the same plan.
- The pool provides default processing parameters for transactions that do not have an associated RCT entry. Such transactions are processed according to the specifications of the TYPE=POOL macro; transactions specified with the TYPE=ENTRY form of the macro are processed according to specifications of that macro, even if they overflow to the pool.
- The THRDM and THRDA parameters must be specified as greater than or equal to 3 for the TYPE=POOL form of the macro.
- All transactions that do not have an RCT entry must be bound with the plan specified for the pool, unless dynamic plan allocation is specified for the pool entry.
- The value on the THRDM parameter defaults to 99 if the parameter is coded greater than the allowed maximum (99).

DSNCRCT TYPE=ENTRY

The TYPE=ENTRY form of the macro defines overrides to values set in the TYPE=POOL definition and provides dedicated threads to a transaction or group of transactions.

The plan name identifies the DB2 resource being accessed by a CICS application. A dynamic plan exit program is a way of allocating plans to CICS dynamically at execution time. (See Appendixes (Volume 2) of *Administration Guide* for more information on dynamic plan exit programs.) In entry threads, all the transactions assigned to the same RCT entry use the plan name or dynamic plan allocation exit program specified for that entry.

Allocation deadlocks might occur if more than one thread is allowed for an RCT entry that uses a plan containing exclusive locking (table space locks). (See the THRDA and TWAIT parameters explanations for more information.)



Notes:

- The macro name must be followed by one or more blanks before optional parameters are coded.
- Multiple parameters must be separated by commas (with no blanks).
- Code a non-blank character in column 72 to indicate that the current statement is continued on the next line.
- RCT macro continuations should begin in column 16.

Description

AUTH=(*identifier*)

Specifies an explicit character string (authorization ID) or directs the attachment facility to use names associated with the CICS transaction.

If you do not specify an explicit character string, you can specify up to three positional subparameters. The position of the positional subparameters indicates the order that the attachment facility selects from transaction-related information to form an authorization ID.

For *identifier*, substitute either an explicit character string or substitute up to three subparameters separated by commas. Subparameters are checked from

left to right until one provides a source from which an authorization ID can be obtained. Permissible values for *identifier* are:

character-string

Specifies a character string that is used as a DB2 authorization ID. For *character-string*, substitute a character string of no more than eight characters, except for the reserved words USERID, USER, TERM, TXID, and SIGNID

If you specify an explicit character string, any subsequent subparameters must be omitted or specified as null (*).

GROUP

Specifies the RACF sign-on user ID (eight character USERID) and the connected group name as the authorization ID. The following table shows how these two values are interpreted by DB2.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF connected group name	If the RACF list of group options is not active, then DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

To use the GROUP option, you must have the following definitions:

- The CICS system must have RACF external security (EXTSEC=YES) specified in the CICS system initialization table (SIT).
- The CICS terminal user's entry in the CICS sign-on table (SNT) must specify EXTSEC=YES.
- If a transaction is started by a CICS command and has no terminal associated with it, the AUTH=GROUP parameter does not apply. (Except for CICS Version 4, which allows a default userid which, in turn, can belong to a secondary group.)
- The CICS terminal user must sign on to CICS.

If no RACF group ID is available for this USERID, then an eight-character field of blanks is passed to DB2 as the group ID.

Null specification (*)

Specifies a null value for the second parameter, the third subparameter, or both. (A null specification for the first subparameter is invalid.)

If nothing is entered for the second and third subparameters, the asterisk (*) is assumed (unless all three specifications are overridden).

SIGNID

Specifies the CICS system authorization ID (up to eight characters) as the resource authorization ID. This is the value of the SIGNID keyword in the TYPE=INIT macro.

TERM

Specifies the terminal identification (four characters padded to eight) as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started (using a CICS command) and has no terminal associated with it, the AUTH=TERM parameter does not apply.

TXID

Specifies the transaction identification (four characters padded to eight) as the authorization ID.

USER

Specifies the sign-on user operator identification associated with the CICS sign-on facility as the authorization ID (three characters padded to eight). An authorization ID cannot be obtained in this manner if a user is not signed on to the entering terminal.

If an operator must be signed on before accessing the DB2 database, USER must be specified in the first position, and the second and third positions must be specified as null (*). For example:

AUTH=(USER,*,*)

If a transaction is started (using a CICS command) and has no terminal associated with it, the AUTH=USER parameter does not apply unless you are using CICS Version 4.

This option is mutually exclusive with the USERID option specified on the same RCT entry.

USERID

Specifies the sign-on user ID (eight-character USERID) associated with the CICS sign-on facility as the authorization ID. An authorization ID cannot be obtained in this manner if a user is not signed on to the entering terminal.

If an operator must be signed on before accessing the DB2 database, USERID must be specified in the first position, and the second and third positions must be specified as null (*). For example:

AUTH=(USERID,*,*)

If a transaction is started (using a CICS command) and has no terminal associated with it, the AUTH=USERID parameter does not apply unless you are using CICS Version 4.

This option is mutually exclusive with the USER option specified on the same RCT entry.

When the sample sign-on exit DSN3SSGN is used with AUTH=USERID, the exit sends the user ID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used and the RACF LISTOFGROUPS option is active, AUTH=USERID acts the same as AUTH=GROUP.

#

Default: (USER,TERM,TXID)

Causes the CICS attachment facility to first try using the CICS sign-on identification as an authorization ID. If the terminal operator is not signed on, the terminal identification is used. If the transaction is not connected with a terminal, the transaction identification is used as an authorization ID.

DPMODE=HIGH|EQ|LOW

Specifies the priority of thread subtasks relative to CICS main task priority.

HIGH

Specifies that subtasks can attain a higher priority than the CICS main task from which the subtask was generated. Use this option for high priority and high volume transactions.

EQ

Specifies that CICS must allow for subtasks to attain equal priority.

LOW

Specifies that subtasks have a lower priority than the CICS main task priority.

PLNEXIT=NO|YES

Specifies whether a dynamic plan allocation exit program is invoked. For more information on dynamic plan allocation, see Appendix B (Volume 2) of *Administration Guide* .

NO

Specifies that this transaction ID entry does not use a plan exit. The CICS attachment facility obtains the plan name using the name specified on the option PLAN=. Specifying NO means you must code the option PLAN=*plan-name*.

YES

Specifies that this transaction ID can dynamically allocate the plan name when the first SQL statement is processed for the application program. This is accomplished by means of the dynamic plan exit specified on the option PLNPGME=. Specifying YES means you must NOT code the option PLAN=.

Default: NO

PLAN=*plan-name*

Specifies the plan ID to be associated with this transaction when it is different from the transaction ID. For *plan-name*, substitute a valid application plan name.

The PLAN ID specified with a TYPE=ENTRY macro is used even if the POOL provides the thread. The PLAN parameter is a required parameter when the TXID parameter is coded as a list of two or more transaction IDs.

If PLAN= is not specified and PLNEXIT=NO, then TXID is the default plan if only one transaction is specified on the TXID keyword. See the description of the TXID parameter on page 2-416.

PLNPGME=*exit-program-name*

Specifies the name of the exit program this entry uses. The default is set by the PLNPGMI parameter on the TYPE=INIT statement. For information on how to write your own exit program, see Appendixes (Volume 2) of *Administration Guide*.

Default: DSNCEXT

ROLBE=YES|NO

Specifies a disposition for transactions defined by this entry in the event a transaction is selected by DB2 as the victim in a deadlock resolution. The

specification of ROLBE overrides the specification of the ROLBI parameter on the TYPE=INIT macro.

YES

Specifies that the attachment facility is to issue a sync point rollback before returning control to the application. A SQL return code of -911 is returned to the program. Specifying YES provides compatibility with SQL/DS.

NO

Specifies that the attachment facility is not to initiate a rollback for this transaction. A SQL return code of -913 is returned to the application.

Default: YES

TASKREQ=*function-key-ID*

Specifies the transaction identification for this entry when the transaction is started using one of the 3270 function keys. Valid specifications are the same as those supported by the TASKREQ operand in the CICS program control table. The options supported include:

PA1, PA2, or PA3

for the PA keys

PF1 through PF24

for the PF keys

OPID

for the operator identification card reader

LPA

for a light pen detectable field on a 3270 device

MSRE

for the 10/63 character magnetic slot reader.

Refer to the CICS documentation for a more detailed description of possible options.

Either TXID or TASKREQ or both parameters are required.

Whenever TASKREQ is coded, with or without a TXID operand, a transaction list is generated. The connection to DB2 for all transactions in this list is made as specified by the other parameters associated with the entry. The first or only transaction identifier specified in the TXID operand (or the first or only function-key identifier, if no TXID was specified) is placed into the resource control table entry.

Because the hexadecimal identifiers of function-key driven transactions generally consist of undisplayed characters, you could have some difficulty interpreting messages containing them. These identifiers appear in both CICS and DB2 messages.

THRDA=*integer*

Specifies the maximum number of threads that the attachment facility allows connected for this transaction, group, or pool before requests are either made to wait or are diverted to the pool. (See the description of the TWAIT parameter.)

For *integer*, substitute a value that does not exceed 99 or the THRDM value. The general restriction is (THRDS ≤ THRDA ≤ THRDM ≤ 99). When

THRDA=0, TWAIT=YES or TWAIT=POOL causes all threads to be diverted to the pool. Forcing low-use transactions into the pool this way might save MVS ATTACH overhead if pool threads are available. If the entry has any protected threads (THRDS > 0), you cannot modify THRDA to be 0. In other words, protected threads are not possible if THRDA=0.

If a plan specified for an RCT entry has exclusive locking, set the value of the THRDA parameter to 1 and the value of the TWAIT parameter to YES to prevent allocation deadlock.

For considerations using threads in conjunction with dynamic plan selection, see Appendix B (Volume 2) of *Administration Guide*.

Default: 0

THRDM=*integer*

Specifies the maximum number of threads the attachment facility is prepared to connect for this transaction group. For *integer*, substitute an integer value that does not exceed 99. The general restriction is (THRDS ≤ THRDA ≤ THRDM ≤ 99).

The value specified for THRDM is the maximum value that can be specified in the DSNM MODIFY TRANSACTION command when using that command to change the value of THRDA. See Section 4 (Volume 1) of *Administration Guide* for a description of the DSNM MODIFY TRANSACTION command.

You can associate a CICS transaction with a plan name without allocating it to an entry thread by specifying no threads (THRDM=0) and overflow to the pool (TWAIT=POOL). The entry thread uses the plan name associated with its TYPE=ENTRY parameter when it overflows to the pool.

For considerations using threads in conjunction with dynamic plan selection, see Appendix B (Volume 2) of *Administration Guide*.

Default: 0

THRDS=*integer*

Specifies the maximum number of MVS subtasks or threads that must be started when the attachment facility is started. This parameter also specifies the number of protected threads. For *integer*, substitute a value that does not exceed the THRDA value or 99. The general restriction is (THRDS ≤ THRDA ≤ THRDM ≤ 99).

For considerations using threads in conjunction with dynamic plan selection, see Appendixes (Volume 2) of *Administration Guide*.

Default: 0

TOKENE=NO|YES

Specifies whether the CICS attachment facility produces a DB2 accounting record for every CICS transaction, even those transactions that are reusing threads. It also specifies whether the CICS attachment facility passes the CICS LU6.2 (protected) token to DB2 for inclusion in the DB2 accounting trace records. You might receive more than one DB2 accounting record for a CICS transaction that has more than one DB2 unit of recovery, but you can correlate the CICS and DB2 records with the matching LU6.2 tokens.

Because CICS produces accounting records on a transaction basis, and DB2 produces accounting records on a thread basis, it can be difficult to correlate the two. This parameter gives you the option of producing a DB2 accounting

record for each CICS transaction, even for transactions that are reusing threads.

If you do not specify YES or NO for TOKENE, then it assumes the value specified in TOKENI on the TYPE=INIT statement.

YES

Specifies that the CICS attachment facility requests that DB2 (using SIGNON) produce an accounting record after each transaction. It also indicates that the attachment facility passes the CICS LU6.2 token to DB2 for inclusion in the DB2 accounting trace records. Specifying YES makes it easier to correlate DB2 and CICS accounting and trace records.

Specifying YES slightly increases the overhead of a SQL request that reuses threads due to additional SIGNON activity. In a thread reuse situation, the transaction rate can degrade by no more than 5%. For additional information on the CICS task scope and DB2 thread scope, see Section 5 (Volume 2) of *Administration Guide*.

NO

Specifies that the CICS attachment facility does not produce a DB2 accounting record during thread reuse. When TOKENE=NO is specified, it is more difficult to correlate DB2 and CICS accounting and trace records.

TWAIT=YES|NO|POOL

Specifying TWAIT overrides the value of the TWAITI parameter on the TYPE=INIT macro.

YES

Specifies that, if all threads are busy, a transaction must wait until one becomes available. A transaction can wait as long as CICS allows it to wait, generally until a thread becomes available. There is a limit to the number of transactions waiting, which is the value specified by CICS MXT (maximum number of tasks) and CMXT (class maximum task).

If TWAIT=YES is specified with THRDA=0, the attachment facility routes the transaction to the pool. Otherwise, a DB2 transaction could wait indefinitely for a thread.

An alternative to using MAX CLASS (CMXT) is using TWAIT=POOL. The task picks up a pool thread rather than waiting for an entry thread to become available.

NO

Specifies that, if all threads are busy, a transaction must be terminated with an abend. If NO is specified, or if TWAIT=NO has been specified on the TYPE=POOL macro, you must closely coordinate the number of threads specified with the MAX CLASS (CMXT) of the transactions. This helps to prevent abends when threads are unavailable.

POOL

Specifies that, if all threads are busy, a transaction must be diverted to use the pool of threads. If the pool is also busy, and NO has been specified for the TWAIT parameter on the TYPE=POOL form of the macro, a transaction is terminated with an abend. See the description of the TWAIT=NO parameter.

Default: YES

TXID=(transaction-ID)

Specify the transaction identification, or identifications for this entry. For *transaction ID*, substitute the transaction identifications found the CSD transaction definition. The way you code this option depends on how many transactions you have and on whether: (1) you have different plans for each transaction; (2) you want to use dynamic plan allocation; (3) you want separate statistics for each transaction.

1. If you have several transactions that use the same plan, you can code a list of transaction IDs, to be indexed to the same RCT entry. Code your entry as in this example:

```
DSNCRCT TYPE=ENTRY, TXID=(TXI1, TXI2, TXIn), PLAN=(PLNA)
```

You cannot code more than one plan per entry. To specify a different plan for each transaction, you must code a separate DSNCRCT entry for each plan, as follows:

```
DSNCRCT TYPE=ENTRY, TXID=(TXI1), PLAN=(PLNA)
DSNCRCT TYPE=ENTRY, TXID=(TXI2), PLAN=(PLNB)
DSNCRCT TYPE=ENTRY, TXID=(TXIn), PLAN=(PLNC)
```

2. With dynamic plan selection, DB2 selects a plan based on the DBRM of the first SQL statement, or based on a user-defined exit routine for dynamic plan selection. You can use the PLNEXIT and PLNPGME parameters to specify a user-defined exit routine.

To use dynamic plan selection for your transactions, code one or more transactions per entry, and optionally add pointers to the PLNEXIT and PLNPGME parameters, as follows:

```
DSNCRCT TYPE=ENTRY, TXID=(TXI1, TXI2, TXIn), PLNEXIT=YES
```

or

```
DSNCRCT TYPE=ENTRY, TXID=(TXI1), PLNEXIT=YES
DSNCRCT TYPE=ENTRY, TXID=(TXI2), PLNEXIT=YES
DSNCRCT TYPE=ENTRY, TXID=(TXI3), PLNEXIT=YES
```

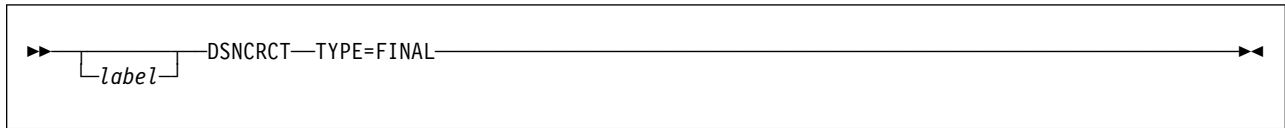
3. If you want separate CICS attachment facility statistics for each transaction, you must code a separate entry for each transaction, as in this example:

```
DSNCRCT TYPE=ENTRY, TXID=(TXI1)
DSNCRCT TYPE=ENTRY, TXID=(TXI2)
DSNCRCT TYPE=ENTRY, TXID=(TXIn)
```

There are two types of entry threads: protected and unprotected. If a thread is protected (THRDS>0), the thread is terminated if unused after two consecutive purge cycles. After a MVS subtask is attached, that MVS subtask remains available until the attachment facility is stopped or the number of active MVS subtasks reaches THRDMAX-2. When the thread is terminated, the associated MVS subtask is not detached. Threads remain assigned to transactions regardless of whether they are driven by a terminal until EOT, regardless of any SYNCPOINT processing.

DSNCRCT TYPE=FINAL

The TYPE=FINAL form of the macro is coded last and results in the generation of the resource control table.



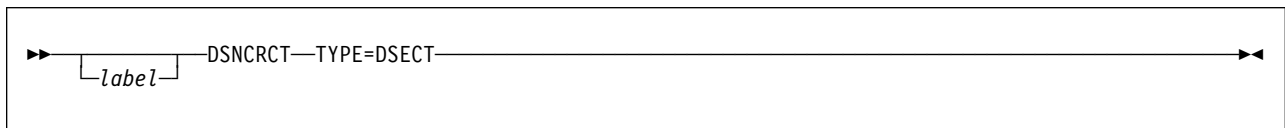
Description

DSNCRCT TYPE=FINAL

This is the name of the macro. It must be coded exactly as it appears here. The TYPE=FINAL form of the macro has no other parameters.

DSNCRCT TYPE=DSECT

The TYPE=DSECT form of the macro is used to generate a DSECT map of the resource control table. It is the only form of the macro that is required to generate the map.



Description

DSNCRCT TYPE=DSECT

This is the name of the macro. It must be coded exactly as it appears here. The TYPE=DSECT form of the macro has no other parameters.

Section 3. Communicating with Other Systems

Chapter 3-1. Connecting Distributed Database Systems	3-3
Installing the DB2 Subsystem	3-4
How the Communications Database (CDB) Handles Requests	3-5
Support for Asynchronous Transfer Mode (ATM) Networks	3-6
DB2 and the DCE Directory	3-6
Adding a database object for DB2 to the DCE directory	3-6
Chapter 3-2. Connecting Systems with VTAM	3-9
Step 1: Install VTAM Version 3.4.2 or Version 4.1	3-11
Step 2: Choose Names and a Password	3-11
Names for the Local Subsystem	3-11
A Password for the Local Subsystem	3-12
Names You Need from the Remote Systems	3-12
Names Chosen by Spiffy Computer Company	3-13
Step 3: Define the DB2 Subsystem to VTAM	3-13
The APPL Statement	3-14
The MODEENT Macro	3-18
Step 4: Populate the Communications Database	3-20
SYSIBM.LOCATIONS Table	3-21
SYSIBM.LUNAMES Table	3-22
SYSIBM.USERNAMES Table	3-23
Step 5: Start VTAM to Use DB2	3-24
Step 6: Tune the System	3-24
Controlling Buffer Storage	3-25
Controlling Pacing	3-26
Modifying Default Session Limits	3-28
Modifying Class of Service	3-28
Associating Applications with Modes	3-29
Updating CDB Values	3-32
Calculating Session Limits	3-33
Calculating Session Limits for DB2 Private Protocol Access	3-33
Considerations for Mixed Applications	3-34
Calculating VTAM I/O Buffer Pool (IOBUF) Storage	3-36
Interpreting CNOS Messages	3-37
Sample VTAM Definitions to Connect Two DB2s	3-39
Basic Definitions	3-39
Channel-Connected DB2s	3-42
NCP-Connected DB2s	3-43
Using the Change Log Inventory Utility to Update the BSDS	3-46
Chapter 3-3. Connecting Systems with TCP/IP	3-49
Enabling TCP/IP communication	3-50
TCP/IP Limitations	3-51
Using Two-phase Commit	3-51
Step 1: Install OpenEdition MVS and TCP/IP and Prepare the LE/370 Runtime Library	3-52
Step 2: Enable DDF for OpenEdition	3-52
Step 3: Define the DB2 Subsystem to TCP/IP	3-53
Customize the TCP/IP Data Sets or Files	3-54
Modify the Change Log Inventory Job	3-55

Step 4: Populate the Communications Database	3-56
SYSIBM.LOCATIONS Table	3-56
SYSIBM.IPNAMES Table	3-57
SYSIBM.USERNAMES Table	3-58
Step 5: Start TCP/IP support	3-58
Step 6: Tuning TCP/IP	3-59

Chapter 3-1. Connecting Distributed Database Systems

You can use the distributed data facility (DDF) of DB2 to access data held by other data management systems, or to make your DB2 data accessible to other systems. DB2 does not place any upper limit on the number of systems it can connect to; available storage is the limiting factor.

This section tells you how to set up DB2 using the Virtual Telecommunication Access Method (VTAM) or Transmission Control Protocol/Internet Protocol (TCP/IP) for distributed communications. The information in this section includes:

- “Chapter 3-2. Connecting Systems with VTAM” on page 3-9, which discusses how to set up DB2 and VTAM for remote communication.
- “Chapter 3-3. Connecting Systems with TCP/IP” on page 3-49, which discusses how to use DRDA in a TCP/IP network.

See *Distributed Relational Database Architecture: Connectivity Guide* for instructions on using DB2 with specific data management systems. See *Data Sharing: Planning and Administration* for information about connecting data base management systems (DBMSs) with a data sharing group.

Some of the DRDA enhancements include:

- TCP/IP support with two-phase commit
- DRDA security enhancements

This includes the following security mechanisms:

- USERID only
- USERID and password or USERID and RACF PassTickets
- DCE security for authentication (server support only)
- USERID, password, and new password (password management)
- Stored procedure enhancements
 - Return multi-row answer sets
 - Commit on return
- Bind options
 - REOPT
 - DEFER
 - KEEP DYNAMIC
- Work load balancing

Applications still use two access methods to control remote access:

- The application can direct a query to another DB2 subsystem by using an alias or a three-part name. We call this method *DB2 private protocol access* because DB2 determines where the location is (from the object name in the query) and connects to that subsystem. DB2 private protocol access allows queries only between DB2 for OS/390 subsystems. DB2 private protocol access uses communications connections that are specific to DB2. (In other words, these are *private* connections.)
- The application can connect to another location by using an SQL statement, and can execute packages that have been previously bound at that location (the *server*). We call this method *DRDA access* because the application connects explicitly to a given location. DRDA access allows queries to other database management systems, as well as DB2.

Also, with DRDA access, queries can originate from any system or application that issues SQL statements as an *application requester* in the formats required by DRDA.

Although use of DRDA is not visible to you, information about it is available in *Distributed Relational Database Architecture Reference*. For two-phase commit using SNA, DB2 uses the presumed abort and presumed nothing protocols. If you are using TCP/IP, DB2 uses the sync point manager defined in the documentation for DRDA Level 3. Again, this is not visible to you, but information about presume nothing protocols is contained in *SNA LU 6.2 Peer Protocols Reference*.

DDF uses TCP/IP or VTAM to communicate with other systems. As shown in Figure 100, communication between DB2 and another system uses a set of communications protocols called LU6.2 or TCP/IP. DB2's DDF is really a transaction program that uses LU6.2 protocols to communicate with database management systems or applications that support DRDA, or listens for all requests for the DB2 port number that uses TCP/IP protocols.

Figure 100. Overview of DDF Environment. DB2's DDF uses LU 6.2 or TCP/IP

Setting up a network for use by database management systems requires knowledge of both database management and communications. Thus, you must put together a team of people with those skills to plan and implement the network.

Installing the DB2 Subsystem

The installation options for DDF are discussed in “Distributed Data Facility: DSNTIPR” on page 2-201 and “Distributed Data Facility Panel: DSNTIP5” on page 2-205. Among other things, these installation panels discuss options for DDF startup, VTAM connection, TCP/IP connection, resource limit facility, and the sample test location. Remember that even when you are using only TCP/IP to communicate with remote sites, you must define VTAM to DB2 because DB2 as a requester using TCP/IP protocols generates the unit of work using NETID and LUNAME.

If you have migrated from DB2 for MVS/ESA Version 4 the data in your Communications Database (CDB) has been copied to the DB2 for OS/390 Version 5 CDB by the job DSNTIJSG. This puts the CDB tables in the proper format for the DB2 catalog. The table names in the CDB have changed and new tables have been added in support of TCP/IP.

If this DB2 subsystem services requesters that support extended dynamic SQL, such as SQL/DS, set the DESCSTAT subsystem parameter to YES in the DSNTIJUZ installation job. This allows applications from the requesting system to execute SQL DESCRIBE statements that appear as extended dynamic SQL statements in the requesting system, but appear as static SQL in the DB2 package. The package must be bound while DESCSTAT=YES.

You should test systems with each other to ensure that their communications setups are correct. The Spiffy Computer Company is a hypothetical enterprise we use throughout this chapter to show examples of how to choose certain options. Spiffy needs to test USIBMSTODB21 with USIBMSTODB22, so they must enter USIBMSTODB22 into the sample test location field of the installation panel. For actual testing, USIBMSTODB22 must have DDF installed and active and must have

Support for Asynchronous Transfer Mode (ATM) Networks

Asynchronous Transfer Mode (ATM) is a rapidly-emerging industry standard for networking technology. ATM offers substantial benefits:

- ATM is an open solution with broad vendor support.
- ATM operates today at 155 Mb/second with projections of over 1 Gb/second in the near future. These speeds make large-scale, multimedia computing closer than ever.
- ATM lets network applications reserve network bandwidth. Large, reserved bandwidth is essential for applications such as video on demand.

You can use ATM technology as the underlying transport for either SNA or TCP/IP connections to DB2.

DB2 and the DCE Directory

The Distributed Computing Environment (DCE) is a set of technology components that provide users and applications with transparent access to data, resources, and services located anywhere on a heterogeneous network. The DCE directory service supports access to a central repository that uniquely identifies all resources in a distributed environment. With this directory, users can identify resources with meaningful names so that they do not need to know the locations of resources in the network.

DRDA application requesters, like DB2 Connect (formerly called DDCS) for DB2 for common server can locate and bind to a DB2 application server by adding a database directory object to the DCE directory.

When a database administrator of DB2 for common server selects the directory services type (`dir_type`) of *DCE*, that administrator does not have to configure local directories on each system to allow distributed access to data. The DCE directory serves as a global directory for this kind of configuration information.

Adding a database object for DB2 to the DCE directory

Remember, this support applies only if you have selected DCE directory support at the client.

For each target database to be accessed from systems such as DB2 for common server, a DCE **database object** must be created in the DCE directory. A database object has a name and consists of a set of attributes. The name of the database object contains the name of the cell concatenated to the subdirectory name and the name of the database. A cell is the basic unit of configuration and administration in DCE. Here is an example of the format for a database object name:

```
./.../cell_name/dir_name1/dir_name2/OBJ_NAME
```

There are several attributes that can be specified with the database object name. A database object that describes a DB2 for OS/390 server should include the communication protocol attribute which has tokens for network protocols that are supported. For this database object the name of the database locator object should be blank. See *DATABASE 2 Administration Guide for common servers* for more information on database locator objects.

To create an entry in the DCE directory for a DB2 for OS/390 server that supports already-verified clients, such as DB2 for common server systems, assume that *DB2_MVS1* is the location name of the server. This server is a DB2 for OS/390 Version 5 system

The following is an example of a database object with the name */.../sanjose.ibm.com/subsys/database/DB2_MVS1*:

DB_Object_Type:	D (for database object)
DB_Product_Name:	DB2_for_OS/390
DB_Product_Release:	V5R1M00
DB_Native_Database_Name:	DB2_MVS1
DB_Database_Protocol:	DRDA
DB_Authentication:	CLIENT
DB_Communication_Protocol:	APPC;DB2NET;LUND1;DB2DRDA;IBMRDB;SAME
DB_Database_Locator_Name:	
DB_Comment:	

The tokens in the DB_Communication Protocol as listed in the above table are:

APPC the protocol type

DB2NET the network ID of the target to which the object belongs

LUND1 the LU name where the target can be found

DB2DRDA the transaction program name (TPN) representing the object in the LU

IBMRDB the mode name

SAME the type of security; the options are NONE, PROGRAM, or SAME

The directory entry is created with the DCE command, **cdscp** as follows:

```
dce_login principal password -exec cdscp <<EOF

create object ../subsys/database/DB2_MVS1

add object ../subsys/database/DB2_MVS1 DB_Object_Type = D
add object ../subsys/database/DB2_MVS1 DB_Product_Name = DB2_for_OS/390
add object ../subsys/database/DB2_MVS1 DB_Product_Release = V5R1M00
add object ../subsys/database/DB2_MVS1 DB_Native_Database_Name = DB2_MVS1
add object ../subsys/database/DB2_MVS1 DB_Database_Protocol = DRDA
add object ../subsys/database/DB2_MVS1 DB_Authentication = CLIENT
add object ../subsys/database/DB2_MVS1 DB_Communication_Protocol=
  APPC;DB2NET;LUND1;DB2DRDA;IBMRDB;SAME
EOF
```

The first command logs in to DCE. Before any objects can be created, the DCE administrator should create a schema for the database object and must establish the proper authorizations to create and access the database object.

For more information on the DCE directory and creating objects within the directory, see *DATABASE 2 Administration Guide for common servers*.

Chapter 3-2. Connecting Systems with VTAM

This chapter tells you how to set up DB2 and VTAM for remote communication. For information about enabling communication with non-DB2 database management systems, see *Distributed Relational Database Architecture: Connectivity Guide* and the appropriate product publications.

Terminology: The following communications terms are used in this chapter:

Logical unit (LU)

A source of requests entering the network and a receptor of replies from the network. For example, a particular DB2 is an LU.

Session A logical connection between two LUs. Multiple sessions can run on a single physical connection.

Conversation

A dialog that uses a session to transfer information between transaction programs, such as DB2 to DB2. A single session can support multiple conversations, but only one at a time.

To prepare DB2 for communication using the distributed data facility (DDF), we suggest the following steps. You can do steps 1, 2, and 3 after installing DB2. Steps 6 through 8 are optional.

“Step 1: Install VTAM Version 3.4.2 or Version 4.1” on page 3-11

To use the DDF in DB2 Version 5, you need to install VTAM Version 3 Release 4 Modification 2 or VTAM Version 4 Release 1; you cannot start DDF under an earlier version of VTAM.

For more information about planning your network configuration, see *Planning for NetView, NCP, and VTAM*. See *VTAM for MVS/ESA Network Implementation Guide* for information about installing your system.

To make monitoring of the network easier, consider installing NetView, too. For more information, see page 2-257.

“Step 2: Choose Names and a Password” on page 3-11

You need to choose two names for the local DB2 subsystem: a location name and a logical unit name (LU name). A *location name* distinguishes a specific database management system in a network, so applications use this name to direct requests to your local DB2 subsystem. Non-DB2 database management systems use other terms for this name; for consistency, we use the DRDA term, *RDBNAM*, to refer to non-DB2 systems' relational database names.

An *LU name* is the name by which VTAM recognizes this subsystem in the network. You might need to know the LU names of other systems that can request data from the local DB2 subsystem, or you can use a default LU name of eight blanks.

If you plan to request data from other systems, you need the LU names *and* location names for those serving systems. Most of the time, system administrators and operators need to know both names, because they can use both names in various commands, and DB2 uses both names in messages.

In addition to the names mentioned above, you can choose an optional password to validate your local DB2 subsystem to VTAM. If you use at least

#

#

VTAM version 4 release 2 and the MVS system on which DB2 is running is part of an MVS sysplex, you can choose a generic LU name to define a DB2 group to remote locations. For information about using generic resources, see *VTAM for MVS/ESA Network Implementation Guide*.

“Step 3: Define the DB2 Subsystem to VTAM” on page 3-13

In this section, we tell how to use the VTAM APPL statement to make the DB2 subsystem known to VTAM. You must include the APPL definitions in the VTAM SYS1.VTAMLST library at VTAM startup.

We also tell how to use the VTAM MODEENT statement to define default session modes. DB2 uses one default mode for DRDA access conversations and another for DB2 private protocol access conversations. You must include mode tables in the VTAM SYS1.VTAMLIB library at VTAM startup.

Sample VTAM definitions are provided in “Sample VTAM Definitions to Connect Two DB2s” on page 3-39, as well as in examples throughout this chapter.

“Step 4: Populate the Communications Database” on page 3-20

The DB2 catalog includes the communications database (CDB), which contains several tables that hold information about your connections with remote systems. You must populate some of these tables before you can request data from those remote systems. If this DB2 system only services data requests, you do not have to populate the CDB; you can use the default values.

“Step 5: Start VTAM to Use DB2” on page 3-24

When you start VTAM to use DB2, you must be sure that the proper definitions are in the VTAM libraries VTAMLST and VTAMLIB.

“Step 6: Tune the System” on page 3-24

This is an optional step, which you can do after you have established communications between two or more systems. The procedure outlined up to this point gives you default values for your DB2 modes and your class of service. Although the defaults are probably adequate for your preliminary testing, you can change them to improve performance in the network, or to assign different modes to different application plans. VTAM publications, such as *VTAM for MVS/ESA Network Implementation Guide*, contain more detailed information about tuning the network.

In this section, we discuss session and mode options you can modify. When VTAM links two nodes, it establishes a session. The number of sessions available can have a significant impact on performance; therefore, you might need to modify your session limit values. “Calculating Session Limits” on page 3-33 contains more detailed information about calculating session limits. Also, large amounts of DB2 data travelling through the network can severely affect VTAM storage, and you might need to tune buffer storage. “Calculating VTAM I/O Buffer Pool (IOBUF) Storage” on page 3-36 contains more detailed information about calculating VTAM buffer pool storage.

You can also tune the system by changing mode options. A *mode* describes various characteristics of a session, such as the maximum number of bytes sent at one time. Modes can point to a *class of service* table, which ranks the available virtual routes for this mode with respect to preference of use and paths through the network. Essentially, the class of service table allows you to assign different network priorities to your modes.

Step 7: Create Aliases

This is an optional step. Each DB2 location can create aliases for the tables it wants to access, using DB2 private protocol access, at the other DB2 locations. For more information about this, see Section 2 (Volume 1) of *Administration Guide*.

Step 8: Provide Authorization for an Appropriate Level of Security

See Section 3 (Volume 1) of *Administration Guide* for information about security considerations for distributed data processing.

Step 1: Install VTAM Version 3.4.2 or Version 4.1

If you have already installed or migrated to VTAM Version 3 Release 4 Modification 2 or Version 4 Release 1, you do not need to reinstall. However, you probably need to customize VTAM to work optimally with DB2. Before you install or customize VTAM, you must consider the communication needs of your DB2 connections. Because you could allow your DB2 subsystem to send large amounts of data through the network, you need to reexamine the capacity of your existing network. In some cases, portions of your existing network might need additional communication hardware to provide the required capacity. VTAM publications, including *VTAM for MVS/ESA Network Implementation Guide* and others, contain more information about these considerations.

Step 2: Choose Names and a Password

In this step, you choose names for your local DB2 subsystem, and, possibly, a VTAM password for it. We also describe the conditions under which you need to know the names of remote systems in the network.

Names for the Local Subsystem

You define the names for the local subsystem and its VTAM password to DB2 by using the installation panels, or by using the change log inventory utility as described in “Using the Change Log Inventory Utility to Update the BSDS” on page 3-46. Choose the following names for the local DB2 subsystem:

- A unique name by which the other systems in the network can recognize your subsystem. The name can have from 1 to 16 characters and is called the *location name*. (This name is the relational database name (DRDA RDBNAM) used in non-DB2 SAA database management systems.) Make sure that the local location name is different from the name of every other system in the network, no matter where it is physically located.

You must share the location name with the other systems that need to send SQL requests to this one.

The location name should not change even if the network changes. Therefore, the allocation of location names should be subject to tight control. To ensure uniqueness, we recommend that you use an IBM-registered SNA NETID as the first six bytes of your location name. If location names are not unique, you have to change many programs and tables if your network is later joined with another network using the same location name.

The IBM recommendation for the NETID is the following format:

- The first two bytes are the country code as defined in ISO standard ISO 3166. These codes include the uppercase letters A through Z.
- The next four bytes are the enterprise code of the registering enterprise. This might already be registered with IBM as your SNA NETID. The enterprise code can include the uppercase letters A through Z, the numbers 0 through 9, and the underscore character (_).

To register your SNA NETID, see your IBM representative.

- A name by which VTAM can recognize the local subsystem. It must be either a unique name or in some cases a generic name.
 - The *unique* name must be unique within the network of connected systems, can have from 1 to 8 characters, and is called the *LU name*. The LU name and the location name of a subsystem can be identical, but we do not recommend this; LU names are unique only within a network, and networks can change. You must share the LU name with any system that requests data from your local subsystem. Later, you enter this name in the VTAM APPL statement described in “Step 3: Define the DB2 Subsystem to VTAM” on page 3-13.
 - If you use VTAM Version 4 Release 2, and the MVS system on which DB2 is running is part of an MVS sysplex, you can use a *generic* 8-character name to represent a group of VTAM LU names. Specify the generic LU name in the field DB2 GENERIC LUNAME on installation panel DSNTIPR. Use column GENERIC of SYSIBM.LUNAMES to indicate that you want to use the generic LU name for CNOS processing and SQL requests to a particular server. A generic LU name is most useful in a data sharing environment. See *Data Sharing: Planning and Administration* for instructions on setting up a generic LU name for a data sharing group. Or, if your network is in a transitional period, consider using generic names to reference network nodes.

A Password for the Local Subsystem

Choosing a VTAM password is optional but recommended. It can have from one to eight EBCDIC characters. If you decide to use a password, you must enter it on the PRTCT option of the VTAM APPL statement. This password is not transmitted through the network, so there is no need to share the password with the other systems. For more information about this password, see Section 3 (Volume 1) of *Administration Guide*.

DB2 does not require you to use a password as long as you have not included one in the VTAM APPL statement.

Names You Need from the Remote Systems

When setting up the communications database (CDB) in the local DB2, you have to know the location names (or DRDA RDBNAMs) and LU names of only remote systems from which you want to request data. The local DB2 does not need location names of systems that request its data; however, you need to know those systems' LU names if you intend to change default communication options.

DB2 does not receive the DRDA RDBNAM from non-DB2 requesters and thus, in those cases, displays only LU names in messages, display output, and trace

output. To help you distinguish between location names and LU names in those cases, the LU name is enclosed in less-than (<) and greater-than (>) brackets.

When your systems begin communicating, you and others involved in working with distributed systems need to be aware of the LU name to DRDA RDBNAM mappings. When you have obtained the necessary names, enter them in the CDB as described in “Step 4: Populate the Communications Database” on page 3-20.

TPNs: Some non-DB2 servers have an additional name that uniquely identifies those servers. In LU 6.2, this is known as a *transaction program name* (TPN), and can be from 1 to 64 characters long. When a DB2 subsystem communicates with other DB2 subsystems, you do not need to supply TPN values. The DB2 subsystems automatically choose the correct TPN values for both DB2 private protocol access and DRDA access.

You might need to supply TPN values when a DB2 subsystem requests data from a non-DB2 system. For cases where the non-DB2 system does not accept the default TPN for DRDA access, enter into your CDB the TPN chosen by that system. When the non-DB2 system uses DRDA access to request data from a DB2 server, the non-DB2 system must use either the TPN name X'07F6C4C2' or DB2DRDA, which are the only values DB2 recognizes when it accepts a request from another system. Some non-DB2 systems enter the TPN as two separate fields: a 1-byte prefix (X'07') and a 3-byte suffix ('6DB').

The TPN is necessary when you are communicating with SQL/DS databases. With SQL/DS, you can have many database management systems running on the same system. Thus, VTAM needs a way to choose the specific database to connect to. For SQL/DS, the TPN is the SQL database machine ID.

The TPN might be necessary when you are communicating with other databases, such as DB2 Common Server Version 2.1.1, or later releases. This is to identify the server with which you are communicating on a particular system.

Names Chosen by Spiffy Computer Company

Spiffy has chosen the location names and LU names shown in Table 120, some of which we use in later examples.

Table 120. Spiffy's Location Names, LU Names, and Transaction Program Names (TPNs)

Location Name	LU Name	TPN	Comments
USIBMSTODB21	LUDB21		DB2 *
USIBMSTODB22	LUDB22		DB2
USIBMSTOSQL1	LUSQLDS	TPNSQLDS1	SQL/DS production system
USIBMSTOSQL2	LUSQLDS	TPNSQLDS2	SQL/DS test system

Note: USIBMSTODB21 plans to accept requests from many OS/2 requesters.

Step 3: Define the DB2 Subsystem to VTAM

You need to use the following VTAM objects in this step:

- An APPL definition statement, described in “The APPL Statement” on page 3-14
- A MODEENT macro, described in “The MODEENT Macro” on page 3-18.

The APPL Statement

A VTAM APPL definition statement defines the VTAM options for the DB2 subsystem and includes it in a major node. Spiffy uses the statement in Figure 102 for the USIBMSTODB21 DB2 subsystem:

```
LUDB21 APPL    APPC=YES,                X
                ATNLOSS=ALL,            X
                AUTH=(ACQ),             X
                AUTOSES=1,              X
                DMINWNL=25,             X
                DMINWNR=25,            X
                DSESLIM=50,            X
                MODETAB=DB2MODES,      X
                PRTCT=D02DN,           X
                SECACPT=ALREADYV,      X
                SRBEXIT=YES,           X
                SYNCLVL=SYNCPT,        X
                VERIFY=NONE,           X
                VPACING=2
```

Figure 102. Example of a VTAM APPL Definition Statement

The sections that follow describe the APPL options that Spiffy uses and a few more in which you might be interested. There are others you can use; for information about those, see *VTAM for MVS/ESA Resource Definition Reference*.

Options For Which You Must Choose Values

For some options, you must supply a specific value; for others, DB2 suggests values that are not the VTAM defaults. In your APPL statement, you must code values for the following:

name The 1 to 8 character LU name you chose in “Step 2: Choose Names and a Password” on page 3-11. For their USIBMSTODB21 DB2 system, Spiffy uses LUDB21.

AUTOSES The number of contention winner sessions that VTAM is to activate automatically between this DB2 and another system on a given mode before DB2 requests a conversation to be created.

Contention occurs when two LUs want to allocate a conversation at the same time in the same session. In order to resolve contention situations, VTAM denotes one LU as the contention winner and one as the contention loser. The winner automatically prevails and is allowed to allocate its conversation. The loser must wait to allocate its conversation.

The default is 0. The suggested value is 1 or greater to ensure that VTAM informs DB2 if a session is inactivated.

Too large a number can take up storage and create resources that are not used. A small number can result in a one-time delay to bring up additional sessions when they are needed by an application.

DMINWNL The minimum number of parallel sessions in which, if there is contention for a conversation, this local DB2 subsystem is the winner.

The suggested value is one-half the value of DSESLIM, described below.

DMINWNR For the same situation as described for DMINWNL, the number of sessions in which the remote system is the winner. The suggested value is one-half the value of DSESLIM, described below.

For more information about session negotiations, see “Interpreting CNOS Messages” on page 3-37.

DSESLIM The default maximum number of sessions allowed for this DB2 subsystem as it communicates with any other system on a given mode. For performance reasons, the DB2 suggested value for DSESLIM is the maximum number of sessions that can possibly be in use on any mode. For example, assume you have 5 modes for which the following maximum numbers of sessions could be active: 10, 12, 20, 30, 40. In this case, DSESLIM should be 40.

Because calculating a precise value for this number can be rather difficult if you do not know exactly how many applications run on a specific mode, Spiffy chooses 50. They can modify this option later if they have problems obtaining enough sessions, or if they find they are requesting sessions that they never need. For information about how to calculate this number, see “Calculating Session Limits” on page 3-33.

You can use DSESLIM to control the number of sessions that this subsystem can issue or receive. For example, to avoid overloading this subsystem with requests from remote application processes, you can assign a low number to DSESLIM to limit the number of simultaneous remote requests issued by a given partner and mode.

Use the CONVLIMIT column of the LUMODES table in the CDB to override this value for specific cases. See “Update SYSIBM.LUMODES with Conversation Limits” on page 3-30 for information about how to do that.

MODETAB The name of the VTAM logon mode table you use to define DB2 session modes. Only modes defined in this table are eligible for conversations created by the local DB2. If you leave this blank, DB2 uses the default mode table shipped with VTAM (ISTINCLM). Spiffy decides to set up a separate mode table and chooses the name DB2MODES. DB2 cannot use either the default mode table or the one you set up yourself until you make entries into the table as described in “The MODEENT Macro” on page 3-18.

PRTCT If you decided to use a password, as described in “Step 2: Choose Names and a Password” on page 3-11, this is that password. Later, you must store the same password in the bootstrap dataset (BSDS), entering it through installation panels or the change log inventory utility.

If you prefer not to use a password, omit this option. The installation panels and the change log inventory utility do not require you to enter a password. For more information about using the VTAM password, see Section 3 (Volume 1) of *Administration Guide*.

SECACPT The level of conversation-level security allowed. We recommend ALREADYV, which gives you the most flexibility in determining your security. You can use the CDB to determine levels of security on a

more granular basis as described in Section 3 (Volume 1) of *Administration Guide*.

We do not recommend SECACPT=CONV because in many cases, it does not allow already verified conversations for DRDA access. It works for conversations that use only DB2 private protocol access.

VERIFY Whether you want SNA partner LU verification. The default, VERIFY=NONE, means that any system can connect with yours. Because Spiffy is setting up a small, restricted network, it chooses the default for now.

Use VERIFY=REQUIRED to activate partner LU verification. This means that you let RACF and VTAM check the identity of an LU that is attempting to connect with yours. For partner LU verification, you must have RACF 1.9 or its equivalent. For more information about partner LU verification, see Section 3 (Volume 1) of *Administration Guide* and *VTAM for MVS/ESA Network Implementation Guide*.

DB2 has no dependency on the value you choose.

VPACING The maximum number of messages that another system can send to this local DB2 subsystem during a conversation before waiting to receive a pacing response. The suggested value is 2.

These message sizes are determined by the RUSIZES option of the MODEENT macro. VPACING and RUSIZES, together with some overhead, determine the amount of storage required for the pacing window. For more information about pacing, see “Controlling Pacing” on page 3-26.

Options You Must Code Exactly as Given

In some cases, DB2 requires particular values of APPL options. For the following, you must code the options exactly as shown; they are *not* the VTAM defaults:

APPC=YES Tells VTAM that DB2 uses APPC conversation verbs.

ATNLOSS=ALL Causes VTAM to notify DB2 each time an SNA session terminates. This is important for timely resource recovery in the two-phase commit process.

AUTH=(ACQ) Determines the DB2 system authority to use certain VTAM functions.

SRBEXIT=YES Tells VTAM that DB2 uses service request block (SRB) processing in its exit routines.

SYNCLVL=SYNCPT

Tells VTAM that DB2 supports two-phase commit. Other systems communicating with this DB2 use this indication to determine if DB2 supports the updating of many locations in one unit of work.

Coding SYNCLVL=SYNCPT does not preclude the support of partner LUs that do not support two-phase commit. DB2 still supports the non-two-phase process. See Section 4 of *Application Programming and SQL Guide* for information about writing applications that access partners that do not have two-phase commit processing.

Options that Must Use VTAM Defaults

For the following options, DB2 *must* use the VTAM defaults; you do not need to code the options:

- HAVAIL=NO** Indicates whether an XRF session can be supported. DB2 requires the default, NO.
- PARSESS=YES** Specifies that parallel sessions are allowed. This defaults to YES when APPC=YES.
- ENCR=NONE** Specifies information about specific cryptographic requirements. There is no support for encryption in this release of VTAM for LU 6.2 applications; therefore, this must be NONE.
- SONSCIP=NO** Specifies information about SCIP exit routines. DB2 does not have SCIP exit routines; this must be NO.
- VTAMFRR=NO** Specifies whether a VTAM functional recovery routine is in effect when control is returned to DB2. DB2 uses its own recovery routines; this must be NO.

Other Options of Interest

In most cases, you can reasonably use the VTAM defaults at first, as Spiffy does. You can change them later. We list them here in case you have some reason not to use the default values.

- ACBNAME** The LU name for the DB2 subsystem. If coded, it overrides the value in *name*. If you use this option, the name must be the same as the LU name defined to DB2 in the bootstrap dataset (BSDS).
- DDRAINL** Whether the local DB2 subsystem wants to accept permission to drain its allocation requests if a change-number-of-sessions (CNOS) request is received that specifies that draining is allowed. The suggested value is the default, NALLOW (do not allow draining).
- DRESPL** Whether the local DB2 is responsible for deactivating sessions when it receives a CNOS request specifying the local DB2 as the responsible system. The suggested value is the default, NALLOW (do not be responsible).
- EAS** The approximate number of concurrent sessions for this DB2 subsystem. For performance reasons, it is better to estimate slightly high. The VTAM default is 509.
- LMDENT** The number of entries to be used for a hash table of other systems. The suggested value is the approximate number of other systems in the network. Spiffy decides to use the default value of 19.
- MAXPVT** The maximum additional amount of private area storage that can be used by VTAM within the DDF address space for the session-related control blocks and messages for DB2. Specifying 0 indicates an unbounded amount; this is the VTAM default.
- OPERCNOS** The ability to have a VTAM operator display and set VTAM session limits for a given LUNAME and MODENAME.
- Use ALLOW to enable a VTAM operator to change session limits dynamically without stopping DDF or changing the CONVLIMIT column of the SYSIBM.LUMODES table.

- Use NALLOW, the default, to make sure VTAM operators are not able to change DB2's session limits dynamically.

OPERCNOS is only available with VTAM Version 3 Release 4 or later releases.

Options Ignored by DB2

The following options are not applicable to DB2 as a VTAM application; do not code them in your APPL statement:

ASLENT	MDLENT	POAQNAM	USSTAB
ASLTAB	MDLTAB	SSCPFM	

The MODEENT Macro

A VTAM link between two systems is a *session*. For every session, there must be a defined set of characteristics called a *mode* existing in a VTAM table called a *log mode table*. This is the table you named in the MODETAB option of the APPL statement, described on “The APPL Statement” on page 3-14

You can create your own log mode table, or add mode names to the default mode table, called ISTINCLM, that is shipped with VTAM. If you decide to add your modes to the default mode table (ISTINCLM), you can find that table in SYS1.SAMPLIB.

Spiffy decides to use the DB2 default modes at first, but also to go ahead and set up a separate mode table for modes used by DB2 for distributed data processing. They can then populate this table with additional modes as they are needed.

Default Modes

There are the following default modes:

- SNASVCMG is an optional mode. It is reserved for use by VTAM for CNOS processing (described in “Interpreting CNOS Messages” on page 3-37) and exists in the VTAM default log mode table. Because SNASVCMG is reserved for use by VTAM, do not enter it as a mode name in the CDB. If you have decided to set up a separate mode table for DB2, you can, if you choose, copy the SNASVCMG mode entry into your DB2 mode table, or just use it as it exists in the ISTINCLM mode table. See *VTAM for MVS/ESA Resource Definition Reference* for a description of this mode.
- IBMDB2LM is a recommended mode entry because it is used as a default for DB2 private protocol access (formerly called system-directed access) whenever you do not explicitly assign a mode to a session. It does *not* exist in the default table; to use it as a default you must add it to your mode table.
- IBMRDB is a recommended mode entry because it is used as a default for DRDA access (formerly called application-directed access) whenever you do not explicitly assign a mode to a session. It does *not* exist in the default table; to use it as a default you must add it to your mode table.

Use the MODEENT macro to enter each mode into your mode table. When this table is complete, you must assemble and link-edit it into SYS1.VTAMLIB. See *VTAM for MVS/ESA Resource Definition Reference* for more information about creating mode tables.

Sample Mode Entries

The sample mode entries for IBMDB2LM and IBMRDB contain the following options that are necessary for dependent LUs to request VTAM sessions.

COMPROT	PRIPROT	TSPROF	SECPROT
FMFPROF	PSERVIC	TYPE	

The samples in Figure 103 work for both dependent and independent LUs; however, if you have no dependent LUs, it is not necessary to re-assemble your existing mode table with the above options. See *Distributed Relational Database Architecture: Connectivity Guide* for more information about dependent LUs.

The ENCR option is ignored by LU 6.2 and is thus not included in our samples.

```

DB2MODES MODETAB
IBMDB2LM MODEENT LOGMODE=IBMDB2LM, DB2 DEFAULT MODE FOR SYS-DIR ACC X
        TYPE=0, NEGOTIABLE BIND X
        SSNDPAC=X'02', SECONDARY SEND PACING COUNT X
        SRCVPAC=X'00', SECONDARY RECEIVE PACING COUNT X
        RUSIZES=X'8989', RUSIZES IN-4096 OUT-4096 X
        FMFPROF=X'13', LU6.2 FM PROFILE X
        TSPROF=X'07', LU6.2 TS PROFILE X
        PRIPROT=X'B0', LU6.2 PRIMARY PROTOCOLS X
        SECPROT=X'B0', LU6.2 SECONDARY PROTOCOLS X
        COMPROT=X'50A5', LU6.2 COMMON PROTOCOLS X
        PSERVIC=X'06020000000000000000122F00' LU6.2 LU TYPE
IBMRDB MODEENT LOGMODE=IBMRDB, DB2 DEFAULT MODE FOR APP-DIR ACC X
        TYPE=0, NEGOTIABLE BIND X
        SSNDPAC=X'02', SECONDARY SEND PACING COUNT X
        SRCVPAC=X'00', SECONDARY RECEIVE PACING COUNT X
        RUSIZES=X'8989', RUSIZES IN-4096 OUT-4096 X
        FMFPROF=X'13', LU6.2 FM PROFILE X
        TSPROF=X'07', LU6.2 TS PROFILE X
        PRIPROT=X'B0', LU6.2 PRIMARY PROTOCOLS X
        SECPROT=X'B0', LU6.2 SECONDARY PROTOCOLS X
        COMPROT=X'50A5', LU6.2 COMMON PROTOCOLS X
        PSERVIC=X'06020000000000000000122F00' LU6.2 LU TYPE
        MODEEND
        END
  
```

Figure 103. Sample Mode Entries

MODEENT Options

When considering values for modes, realize that the partner system can choose different values. If the partner has different values, VTAM negotiates the values to limits acceptable to both systems when the session is established for the mode.

The options used in the MODEENT macro have the following meanings.

name The *name* option (IBMDB2LM and IBMRDB in the examples) is optional and has no function in the specification of a logon mode table.

LOGMODE Specifies the logon mode name to be used as a key for the session options in this table entry. This logon mode name corresponds to mode name columns in the CDB.

TYPE	TYPE=0 indicates that DB2 is using a negotiable BIND, which is required for communicating with dependent LUs.
SRCVPAC	Specifies the secondary receive pacing count. The DB2 suggested value is X'00'.
SSNDPAC	Specifies the secondary send pacing count. The DB2 suggested value is any nonzero number. Do not use 0; this turns off pacing, which can result in problems with IOBUF storage.
RUSIZES	Specifies the maximum length of data in bytes that can be sent and received in one <i>request/response unit</i> (RU). It is read as two numbers, each having two hexadecimal digits: the first number for the send amount, the second for the receive amount. The suggested value of X'8989' means that VTAM sends a maximum of 4096 bytes (8×2^9) across <i>at one time</i> , but there is no limit on how much total information can be sent.
FMPROF	This constant specifies the function management profile required for LU 6.2.
TSPROF	This constant specifies the transmission services profile required for LU 6.2.
PRIPROT	This constant specifies the primary LU protocols used in LU 6.2.
SECPROT	This constant specifies the secondary LU protocols used in LU 6.2.
COMPROT	This constant specifies the common LU protocols used in LU 6.2.
PSERVIC	This constant specifies this as an LU type 6.2.

Some of the above options can have a profound effect on performance because of their impact on pacing. For more information about these pacing options, see "Controlling Pacing" on page 3-26.

The ENCR option is ignored by LU 6.2; thus it is not included in the sample above.

Step 4: Populate the Communications Database

If you plan to use DB2 only as a server, you do not need to populate the CDB; default values are used. For example, Spiffy's USIBMSTODB21 subsystem works as a server for many OS/2 requesters. It is not necessary for Spiffy to register all those requesters in DB2's CDB.

However, if you intend to request data, you need to insert one row for each remote system into SYSIBM.LOCATIONS and SYSIBM.LUNAMES. You do not need to populate table SYSIBM.LULIST unless DB2 is acting as a requester of data that resides in a data sharing group. See *Data Sharing: Planning and Administration* for more information. Section 3 (Volume 1) of *Administration Guide* discusses the requirements for the other tables.

After you populate these tables, you can write queries that access data at a remote system. For instructions on sending SQL statements to other systems, see *Application Programming and SQL Guide*. For instructions on granting privileges to users on remote DB2 subsystems, see Section 3 (Volume 1) of *Administration Guide*.

SYSIBM.LOCATIONS Table

The table LOCATIONS has the following purposes:

- When you do an SQL CONNECT, the LOCATION column maps the location name (or DRDA RDBNAM) to the VTAM LU name and, if necessary, the transaction program names (TPNs).
- When your DB2 receives a request from another DB2 site (both DRDA access and DB2 private protocol access), it uses the LOCATION column to validate the requesting site's location name. (Only DB2 sites exchange location names in both directions.) You do not need to populate this table for systems that use only DRDA access and make requests only of your local DB2.

LOCATIONS has the following columns relating to VTAM:

LOCATION CHAR(16)

The unique network location name, or DRDA RDBNAM, assigned to a system, remote or local. You must provide location names for any systems that you request data from. This column is the primary key for this table.

LINKNAME CHAR(8)

Identifies the VTAM attributes associated with this location. For each LINKNAME specified, you must have a row in SYSIBM.LUNAMES whose LUNAME matches the value specified in this column. Because this table is used for outbound requests, you must provide an LUNAME or your requests fail. Do not enter blanks in this column.

TPN VARCHAR(64)

This column is used to enter a transaction program name (TPN) for SNA conversations with non-DB2 systems. You only need to use this column if you are sending or receiving SQL requests from systems using non-default TPNs as described in “Step 2: Choose Names and a Password” on page 3-11.

Spiffy's USIBMSTODB21 location wants a LOCATIONS table that looks like Table 121.

Table 121. Spiffy's LOCATIONS Table

LOCATION	LINKNAME	TPN
USIBMSTODB21	LUDB21	
USIBMSTODB22	LUDB22	
USIBMSTOSQL1	LUSQLDS	TPNSQLDS2
USIBMSTOSQL2	LUSQLDS	TPNSQLDS1

For example, add the second row with this statement:

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME)
VALUES ('USIBMSTODB22', 'LUDB22');
```

A Row for the Local Location: You do not need a row for the local DB2 in the LUNAMES and LOCATIONS tables. For example, Spiffy's USIBMSTODB21 subsystem does not require a row that shows its own LU name and location name. However, for convenience, Spiffy decides to populate one LUNAMES table and one LOCATIONS table and to duplicate them entirely at each location. As a result, each table contains a row for its own LU name or location name.

SYSIBM.LUNAMES Table

LUNAMES defines the security and mode requirements for conversations with other systems. Decisions about how to populate this table depend on how you intend to use DB2:

- If you use this system only as a server, DB2 can use a blank in the LUNAME column as a default. DB2 uses the values in the default row as defaults for LUs that are not explicitly defined in LUNAMES. If you do not have a row with a blank in the LUNAME column, DB2 rejects client connections that do not explicitly state a valid LUNAME. The DSNTIJSJG installation job creates the default row in table SYSIBM.LUNAMES.
- If this DB2 requests data from other systems, you need to provide LU names for those systems.

LUNAMES has the following columns:

LUNAME CHAR(8)

The LU name of the remote system. The default of 8 blanks indicates that this row is used for serving the requests of any system that is not specifically listed in the LUNAMES table. For example, because USIBMSTODB21 acts strictly as a server for many OS/2 requesters, Spiffy leaves the LUNAME column blank for those requesters and uses default values for the entire row.

However, you must provide LU names for any remote system that uses different values from the defaults.

SYSMODENAME CHAR(8)

The mode used to establish system-to-system conversations for DB2 private protocol access. This column is ignored for DRDA access conversations. For now, Spiffy leaves it blank to use the default mode, IBMDB2LM, which they entered in step 3.

SECURITY_IN CHAR(1)

Defines the security options that are accepted by this DB2 subsystem when an SNA client connects to DB2. The default, A, means that an incoming connection request is accepted if it includes any of these:

- A user ID
- A user ID and password
- A user ID and RACF PassTicket
- A DCE security ticket.

SECURITY_OUT CHAR(1)

Defines the security option that is used when local DB2 SQL applications connect to any remote server associated with this LUNAME. The default, A, means that outgoing connection requests contain an authorization ID without a password.

ENCRYPTPSWDS CHAR(1)

For now, Spiffy uses a blank to indicate no encryption of passwords. For more information about using this column for security, see Section 3 (Volume 1) of *Administration Guide*.

MODESELECT CHAR(1)

Determines whether to use the default mode or to choose a mode from the MODESELECT table. Spiffy uses a blank to use the default modes: IBMDB2LM for conversations using DB2 private protocol

access and IBMRDB for conversations using DRDA access. For more information about this column, and other tables in the CDB, see “Associating Applications with Modes” on page 3-29.

USERNAMES CHAR(1)

This column is used for inbound and outbound requests to control authorization ID translation.

Spiffy uses a blank to indicate that no authorization IDs are translated, and also that no passwords are sent to the server. For more information, see Section 3 (Volume 1) of *Administration Guide*.

GENERIC CHAR(1)

A 'Y' in this column indicates that a generic LU name is to be used for CNOS processing and SQL requests sent to the partner LU. A value of 'N' or ' ' indicates that the name specified in the LUNAME column is to be used. See Chapter 4. of *Data Sharing: Planning and Administration* for instructions on setting up a generic LU name.

Spiffy's USIBMSTODB21 location wants a LUNAMES table that looks like Table 122.

Table 122. Spiffy's SYSIBM.LUNAMES Table. The row of blanks is a default row that Spiffy intends to use for OS/2 requesters in its initial testing.

LUNAME	SYSMODENAME	USERSECURITY ¹	ENCRYPTPSWDS	MODESELECT	USERNAMES
LUDB21					
LUDB22					
LUSQLDS					
(blanks)					

Note: ¹USERSECURITY refers to SECURITY_IN AND SECURITY_OUT

Spiffy can use an SQL INSERT statement to add the appropriate rows. For example, they add the LU name for USIBMSTODB22 with this statement:

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
VALUES ('LUDB22');
```

SYSIBM.USERNAMES Table

USERNAMES contains information needed for outbound and inbound ID translation and also for *come from* checking.

TYPE CHAR(1)

Whether the row is for inbound or outbound translation.

AUTHID CHAR(8)

Authorization ID to translate. If blank, it applies to all authorization IDs.

LINKNAME CHAR(8)

Identifies the VTAM network location associated with the row. A blank indicates it applies to all SNA partners. For nonblank values, this value must match the LUNAME value in SYSIBM.LUNAMES

NEWAUTHID CHAR(8)

The translated value of AUTHID.

PASSWORD CHAR(8)

The password to accompany an outbound request for passwords that are not encrypted. This column is ignored if the row is for inbound translation or if passwords are encrypted.

Step 5: Start VTAM to Use DB2

You do not need to code any special VTAM start options to use DB2, but you can tailor start option values for DB2 communications. For more information on start options, see *VTAM for MVS/ESA Resource Definition Reference*.

You must start VTAM before starting DDF. For information on how to start VTAM, see *VTAM for MVS/ESA Network Implementation Guide*.

There are two VTAM libraries that must contain definitions for DB2:

- SYS1.VTAMLST contains the definitions that define DB2 as a VTAM application. “Step 3: Define the DB2 Subsystem to VTAM” on page 3-13 contains more information about these definitions.

You can use the following VTAM command to enable DB2, assuming that the member DB2APPLS contains definitions for DB2:

```
V NET,ACT,ID=DB2APPLS
```

- SYS1.VTAMLIB contains mode table definitions used by DDF. This must be an APF-authorized library, or in a concatenation of APF-authorized libraries. For more information about modes and mode tables, see “The MODEENT Macro” on page 3-18.

Step 6: Tune the System

As you begin testing with DB2’s distributed data facility, you probably need to modify VTAM options and CDB values to handle certain problems. We highly recommend that you consult a VTAM communications expert to tune your network. This section describes, at a fairly high level, some of the things to consider when tuning VTAM for DDF. See also Section 5 (Volume 2) of *Administration Guide* for more information about monitoring and tuning DB2 distributed applications.

In this section we describe:

- “Controlling Buffer Storage” on page 3-25.

By sending large amounts of data through the network, DB2 can cause problems with your VTAM I/O buffer pool.

- “Controlling Pacing” on page 3-26.

You probably need to tune your pacing options if your VTAM buffers become overloaded with data that is sent to this local DB2.

- “Modifying Default Session Limits” on page 3-28

Consider modifying session limits if you have problems obtaining enough sessions to handle your distributed workload efficiently.

- “Modifying Class of Service” on page 3-28.

Specifying a class of service can help you assign priorities to your network applications.

- “Associating Applications with Modes” on page 3-29.

Tuning the system can require that you add new modes to your log mode table so that there is a greater variety of classes of service available for your sessions. This variety allows you to have more flexibility in tuning the system for specific uses. This section tells you how to associate specific sessions with modes.

Before you begin tuning the network, you must understand the relationship between VTAM options and associated values in DB2’s CDB. Table 123 summarizes the relationship.

Table 123. Relationship between DB2’s CDB and VTAM Macros

Macro Name	Option	CDB table.column	Relationship
APPL	<i>name</i>	LOCATIONS.LINKNAME LUNAMES.LUNAME LUMODES.LUNAME MODESELECT.LUNAME USERNAMES.LINKNAME LULIST.LINKNAME	The LU name used in VTAM communication. This name maps 1:1 to the system’s location name in LOCATIONS.
APPL	DSESLIM	LUMODES.CONVLIMIT	CONVLIMIT overrides session limits specified with DSESLIM. Session limit values are used in CNOS processing.
MODEENT	LOGMODE	LUNAMES.MODENAME	MODENAME chooses the mode for the system conversation in DB2 private protocol access.
MODEENT	LOGMODE	LUMODES.MODENAME	LUMODES creates session limits for <i>specific</i> LU name and mode name combinations.
MODEENT	LOGMODE	MODESELECT.MODENAME	MODESELECT maps authorization IDs and plans to specific modes.

Controlling Buffer Storage

VTAM uses buffer pools for control blocks, network traffic data, and channel programs. A shortage of buffer pools, can have an adverse effect on VTAM CPU time, storage consumption, and the ability to serve DB2 requests.

You can monitor VTAM buffer pools using one of the following:

- The VTAM command DISPLAY NET,BFRUSE
- A VTAM trace, obtained by entering the following MVS MODIFY command:

```
F procname,TRACE,TYPE=SMS,ID=VTAMBUF
```

Procname in the command is the VTAM start procedure name. The data is collected by the generalized trace facility (GTF).

DB2 applications can consume a large number of VTAM IOBUF pool buffers, depending on the VTAM options you choose and the volume of data being transmitted between distributed systems. You can estimate the number of IOBUF pool buffers, and the storage they use, by using the formula described in “Calculating VTAM I/O Buffer Pool (IOBUF) Storage” on page 3-36.

The VTAM IOBUF pool is an area of storage that VTAM uses to store messages that are exchanged between network resources. The IOBUF pool is shared among all VTAM resources. When you calculate the IOBUF storage required to satisfy DB2 requirements, keep in mind that the IOBUF pool must have enough space to satisfy requests from other VTAM applications as well, such as TSO, CICS, and IMS.

To prevent shortages of these VTAM buffers (IOBUFs), you can do any of the following:

Increase the Number of IOBUF Buffers

The IOBUF pool definition is one of the VTAM start options. You can enter the IOBUF option from the MVS console, or you can include it at VTAM startup in SYS1.VTAMLST in member ATCSTRxx.

Tuning the IOBUF pool encompasses both base allocation and dynamic expansion values. At installation, you can specify a base allocation for the IOBUF pool (in number of buffers) and a dynamic expansion (in number of buffers). When storage runs short in the buffer pool, VTAM temporarily expands the IOBUF pool by the dynamic expansion value, based on a trigger which you can also specify in VTAM definitions. We recommend that you also set a maximum size for the IOBUF pool size using the *xpanlim* start option for the buffer pool. If you turn off pacing accidentally, *xpanlim* prevents DB2 from causing VTAM to grab unlimited amounts of storage. For more information about allocating buffer pools, see *VTAM for MVS/ESA Network Implementation Guide*.

Decrease the Session Level Pacing Count

Pacing is vital for controlling the potentially large amounts of data that are transferred around the network. See “Controlling Pacing” for more information about modifying pacing values.

Decrease the Number of Concurrent Conversations

You can reduce the number of concurrent conversations by reducing the number of sessions. See “Modifying Default Session Limits” on page 3-28 for more information about how to do this.

Decrease the Request Unit (RU) Size

The RUSIZES option is part of the mode entry statement described in “The MODEENT Macro” on page 3-18.

Because reducing the number of sessions and the RUSIZES value can adversely affect performance, you should first consider increasing IOBUF buffers and decreasing the session pacing count.

Controlling Pacing

Session level pacing is the mechanism by which the receiver of data (DB2 in this case) can control the pace at which the sender sends data (in the form of RUs). The pacing size is the number of RUs VTAM sends across the line at one time, and you set that value using the VPACING option of the VTAM APPL definition statement described in “The APPL Statement” on page 3-14. You set the RU size in the MODEENT macro described in “The MODEENT Macro” on page 3-18. The receiving VTAM stores these RUs in its IOBUF pool; it uses pacing so that its buffers do not become flooded with data.

The pacing process works as shown in Figure 104 on page 3-27. The system at the sending side (let's assume it is USIBMSTODB22) passes data to its VTAM. VTAM formats the data into RUs, and sends those RUs across the network. If, for example, the pacing size is 2, then it sends two RUs. A 29-byte network header is sent with each RU.

After USIBMSTODB22's VTAM sends the specified number of RUs, it does not send any more data on this session until it receives a pacing response from the VTAM at USIBMSTODB21. USIBMSTODB21's VTAM does not send a response until VTAM transfers the data into DB2's buffers.

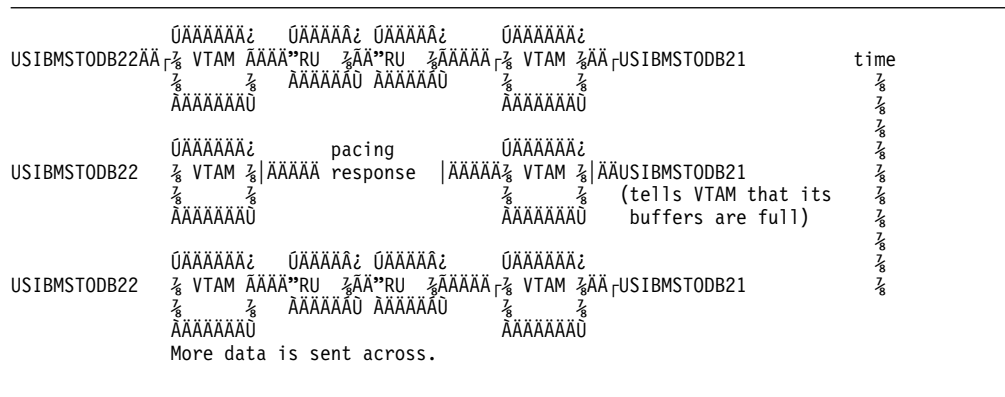


Figure 104. How Pacing Works

Although it is generally true that the receiving system can control inbound pacing, both communicating systems negotiate final pacing values. For more information about pacing negotiation, see *VTAM for MVS/ESA Network Implementation Guide*.

Recommendation for APPL Pacing Option

The VPACING option of the APPL statement is the maximum number of RUs that another LU can send, on a session, to this LU before waiting to receive a pacing response. This should always be a nonzero value, or else you turn off all pacing for all sessions affected by this option.

We recommend starting with a value of 2 for both communicating systems. This pacing size is the same in both directions for all modes.

The VPACING value is used with the RUSIZES option of the MODEENT macro to control the pacing window size. Thus, if VPACING is 2 and RUSIZES is 4KB (X'8989'), then about $2 \times 4KB = 8KB$ are sent before waiting to receive a pacing response. You should verify that your VTAM buffer pools are large enough to accommodate the chosen pacing and RU sizes. See "Calculating VTAM I/O Buffer Pool (IOBUF) Storage" on page 3-36 for information about calculating buffer pool sizes.

Recommendation for MODEENT Pacing Options

The MODEENT macro described on 3-18, contains several pacing options:

PSNDPAC This does not apply to DB2; therefore, you can ignore this option.

SSNDPAC This option is really a flag that you set to either 0 (off) or nonzero (on). When 0, outbound pacing for sessions is disabled, which can lead to severe problems with IOBUF storage. We recommend that you specify a nonzero value for this option.

SRCVPAC If 0, which is the recommended value, the VPACING option of the VTAM APPL statement controls both the send and receive pacing for all sessions in all modes. A value of 0 makes it easier for you to predict pacing results and makes it easier to maintain your pacing definitions.

If nonzero, VPACING controls pacing in one direction, and SRCVPAC controls it in another. LU 6.2 protocols make it difficult to predict which option is in control at any given time.

Modifying Default Session Limits

An understanding of session limits helps you control resource use. DRDA access uses one session per partner. A given application can connect to many partners at any time.

DB2 private protocol access can take advantage of more sessions. For best performance, every read-only cursor in an application can use its own conversation. However, there can be resource constraints that disallow so many sessions. When conversation limits have been reached, DB2 begins sharing available conversations, if the application already owns one or more VTAM conversations. This means that, if an application has acquired its first conversation, it is not rejected because of a shortage of conversations.

As you begin increasing the number of applications that use the distributed data facility, you might find that your applications are waiting for conversations to become available, thus increasing the network delay associated with the application. Therefore, this section also tells how to increase your default maximum session limit to ensure enough resources for best performance.

Increasing Session Limits for Specific Modes and LUs

To fine tune session limits for specific LU name and mode name combinations, you can modify the CONVLIMIT column of the SYSIBM.LUMODES table. To calculate CONVLIMIT, follow up to step 6 in “Calculating Session Limits for DB2 Private Protocol Access” on page 3-33.

If you are running VTAM Version 3 Release 4 or later, and if you have specified OPERCNOS=ALLOW in the VTAM APPL statement, you can change the session limits dynamically with the VTAM command MODIFY VTAM,DEFINE. See *VTAM for MVS/ESA Operation* for more information about VTAM commands.

Increasing Session Limits for All Modes and LUs

You can modify the overall session limit default by modifying the DSESLIM option of the VTAM APPL statement. (Remember: a value in CONVLIMIT for a given LU name and mode name combination still overrides DSESLIM.) The suggested value for DSESLIM is the maximum number of sessions that could possibly be in use on any single mode. The procedure for determining session limits is in “Calculating Session Limits” on page 3-33.

Modifying Class of Service

You can define the transmission priority and paths between systems with entries into a class of service (COS) table. Each entry in the table is associated with a list of routes to be used with a particular class. For example, you might want to place interactive sessions on a faster route than a batch job.

To specify a class of service for a specific mode, use the COS (class of service name) option of the MODEENT macro. When you specify a name of a COS entry in the mode description, you select the list of routes you want to be used for the session. When VTAM establishes the session, it chooses the first available route in the list of routes you tell it to use for that class.

If you do not specify a COS name, the mode gets the default list of routes from VTAM.

Associating Applications with Modes

The information under this heading, up to “Updating CDB Values” on page 3-32 is General-use Programming Interface, as defined in “Notices” on page ix.

As you tune your system, you can assign certain applications, such as a high priority job, to the mode that is best suited for that job. You can also use a specific mode assignment for an application that uses many conversations. You can assign such an application to a mode that allows more conversations than the VTAM DSESLIM value you entered in the APPL statement.

To associate a specific mode with a particular session, you need to update or insert rows into three tables in the CDB: LUNAMES, LUMODES, and MODESELECT.

For information about when updates to the CDB are activated, see “Updating CDB Values” on page 3-32.

Update LUNAMES to Associate Modes with LU Names

This table is described in “Step 4: Populate the Communications Database” on page 3-20. It associates a mode with each remote system that the local subsystem can send a query to. There are two types of connections that you can specify in this table:

- System conversation (used only for DB2 private protocol access)
- SQL processing conversations (can be many for DB2 private protocol access; only one for DRDA access)

Figure 105 shows how these different conversations are used for a DB2 private protocol access application.



Figure 105. Conversations Used for DB2 Private Protocol Access

System Conversation: For DB2 private protocol access, the *system conversation* must be established before any processing can begin. To choose a system mode,

DB2 looks in the MODENAME column of the row in the LUNAMES table that corresponds to the target DB2.

If MODENAME is blank, then IBMDB2LM is used. If MODENAME contains a mode name, then that mode is used when creating the first conversation between the two DB2 subsystems.

SQL Processing Conversations: For DRDA access, an SQL processing conversation is established. The mode name for SQL processing conversations is determined by the MODESELECT table of the CDB. If the MODESELECT column of LUNAMES table is blank or contains N, then the default mode (IBMRDB) is used. If it contains a Y, then MODESELECT is searched.

Spiffy wants to use the mode LOC2MODE for system conversations with USIBMSTODB22's DB2, using DB2 private protocol access. They also want to begin setting up different modes for specific applications to use in conversations with USIBMSTODB22. They enter into their DB2 mode table a mode named LOC2MODE. In a later step, they define LOC2MODE in MODESELECT; for now, they update the LUDB22 row of LUNAMES as shown in Table 124.

Table 124. Spiffy's LUNAMES Table, after Update

LUNAME	SYSMODENAME	USERSECURITY ¹	ENCRYPTPSWDS	MODESELECT	USERNAMES
LUDB22	LOC2MODE			Y	

Note: ¹USERSECURITY represents SECURITY_IN and SECURITY_OUT

Spiffy can use the UPDATE statement below to make the change, which takes effect the next time DDF is started. (It takes effect immediately if DDF is started but USIBMSTODB22 is not yet accessed.)

```
UPDATE SYSIBM.LUNAMES SET MODENAME='LOC2MODE', MODESELECT='Y' WHERE
LUNAME='LUDB22';
```

Update SYSIBM.LUMODES with Conversation Limits

Populating this table is optional; if you do not specify mode names in this table, the VTAM defaults are used. Use this table to provide VTAM with conversation limits for *specific* LU name and mode name combinations. (The table is unlike the DSESLIM option of the VTAM APPL definition statement, which provides the default session limits for *all* LU name and mode name combinations.) The primary key for this table is formed by the LU name and mode name combination. Only one entry with the same LU name and mode name is allowed.

LUMODES is accessed for negotiation of session limits with a remote DB2 for a specific mode. This negotiation is called *change number of sessions* (CNOS), which is discussed in "Interpreting CNOS Messages" on page 3-37.

For example, suppose Spiffy wants to allocate 75 sessions instead of 50 (the value in DSESLIM) for conversations to USIBMSTODB22, using the mode named LOC2MODE. They can use the INSERT statement below to update the value in the CONVLIMIT column to 75. The new session limit takes effect the next time DDF is started, or in the initial connection to this LU for this mode.

```
INSERT INTO SYSIBM.LUMODES VALUES ('LUDB22','LOC2MODE',75,'N');
```


CNOS processing negotiates a value that is the lesser of the number of sessions available at either system for that mode. Therefore, USIBMSTODB22 must also increase its CONVLIMIT value to derive any benefit from the added sessions.

Columns of the LUMODES Table:

LUNAME CHAR(8)

Again, this is the LU name of the other system. This column is a foreign key of the LUNAMES table; thus, all LU names defined in this table must be defined in LUNAMES. When you delete an LU name from the LUNAMES table, all associated rows in LUMODES are deleted.

MODENAME CHAR(8)

The name of the logon mode description in the VTAM logon mode table that VTAM uses when creating a conversation to support the local DB2's request for data from another system. The mode named here must exist in the mode table used by DB2 before a conversation can be created between USIBMSTODB21 and USIBMSTODB22.

CONVLIMIT SMALLINT

The maximum number of conversations to be concurrently active between this DB2 subsystem and the other system for this mode. This number overrides the number in the DSESLIM option of the VTAM APPL definition statement during CNOS processing, as described in "Interpreting CNOS Messages" on page 3-37.

Update SYSIBM.MODESELECT to Associate Plans with Modes

This table maps authorization IDs and plan names to mode names. The primary key for this table is the combination of AUTHID, LUNAME, and PLANNAME. Only one entry with the same AUTHID, LUNAME, and PLANNAME is allowed.

Use this table to make sure that certain authorization IDs using certain plans always have a predefined class of service suited for that operation. For example, the USIBMSTODB21 location might want to work with USIBMSTODB22 to set up a high performance mode for DBADM to run queries to USIBMSTODB22. After the following statement is committed, all subsequent threads to USIBMSTODB22 use mode DB2MODE1 to process SQL processing conversations:

```
INSERT INTO SYSIBM.MODESELECT VALUES ('DBADM',' ','LUDB22','DB2MODE1');
```

Populating this table is optional. If the remaining columns are blank for any given LU name, then the mode name applies to all authorization IDs for all PLANNAMEs accessing the given LU name.

Columns of the MODESELECT Table:

AUTHID CHAR(8)

The authorization ID of the request for data from another system. A blank AUTHID indicates that the specified mode name applies to all authorization IDs. Blank is the default.

PLANNAME CHAR(8)

The plan name associated with the request for data from another system. A blank plan name indicates that the specified mode name applies to all plan names. Blank is the default.

LUNAME CHAR(8)

The LU name to which the specific mode name applies. This column is a foreign key of the LUNAMES table; therefore, all LU names defined in this table must be defined in LUNAMES.

MODENAME CHAR(8)

The name of the logon mode description in the VTAM logon mode table that is used when creating a conversation to support the request for data from another system. If this column is blank, the default mode (IBMDB2LM or IBMRDB) is used.

How an SQL Processing Conversation Mode is Chosen:

The MODESELECT table of the CDB is used to choose a mode for an SQL processing conversation (if the MODESELECT column of the LUNAMES table contains Y for this LU name). Table 125 shows the search order of the MODESELECT table.

Table 125. Precedence Search Order for MODESELECT Table of CDB

AUTHID	PLANNAME	Result
Name	Name	The MODENAME applies to the named AUTHID for the named PLANNAME accessing the named LU.
Name	Blank	The MODENAME applies to the named AUTHID for all PLANNAMEs accessing the named LU.
Blank	Name	The MODENAME applies to all AUTHIDs for the named PLANNAME accessing the named LU.
Blank	Blank	The MODENAME applies to all AUTHIDs for all PLANNAMEs accessing the named LU.

If the MODESELECT column of the LUNAMES table contains Y for a particular LU name and no row is found for that LU name in the MODESELECT table, then you receive a negative SQL return code when trying to access the system at that LU.

Plan Name for Remote Bind Operations: If you want to specify a particular mode for remote bind operations, use the plan name DSNBIND in MODESELECT.

Updating CDB Values

Any table in the CDB can be updated while DDF is active. The changes take effect as follows:

- Changes to LUMODES take effect the next time DDF is started, or on the initial session to a given LUMODE combination.
- Changes to LUNAMES, LOCATIONS, and LULIST take effect as follows:
 - If DDF has not yet tried to communicate with a particular remote location, rows added to LUNAMES and LOCATIONS take effect when DDF attempts to communicate with that location.
 - If DDF has already attempted communication with a particular location, rows added to LUNAMES and LOCATIONS take effect the next time DDF is started.
- Changes to USERNAMES and MODESELECT take effect at the next thread access.

In all cases, existing conversations continue to operate as the table specified before the update.

The process of modifying the CDB, particularly MODESELECT and USER NAMES, can interfere with DDF's access to the tables. This could potentially cause deadlocks and timeouts, which cause the attempted access to the remote system to fail.

Calculating Session Limits

You might have to derive a precise figure for your session limits (DSESLIM). For example, if you specify a very large number for your session limits and you are running short of space, it might help to calculate a number closer to what you actually need. This section tells you how to calculate session limit values based on whether the applications are using DRDA access or DB2 private protocol access.

If you have applications that use both DRDA access and DB2 private protocol access, first read this section, then see "Considerations for Mixed Applications" on page 3-34.

The recommended session limit for a specific mode is the following:

- For DRDA access: the maximum number of concurrently active applications using that mode to and from the remote system.

For example, suppose location A has a maximum of three DRDA access applications that can run concurrently on Mode1 to other locations. Also suppose that a maximum of 10 DRDA access applications can be incoming to location A on Mode1. Thus, Mode1 should support 13 sessions for DRDA access applications.

- For DB2 private protocol access applications: the maximum number of system conversations using that mode name to and from the remote DB2 subsystem, plus the maximum number of conversations needed for each concurrently active application using that mode to and from the remote DB2. The formula for calculating session limits for DB2 private protocol access is outlined in the following section.

Calculating Session Limits for DB2 Private Protocol Access

To calculate the session limits for location A, which uses DB2 private protocol access:

1. Determine all the applications that run concurrently on location A and use Mode1 to access location B. Call these A1, A2, ... AN.
2. For *each* of these applications, determine the total number of read-only cursors that can be concurrently active to location B and add 1. This additional conversation represents the cursor for SQL statements that modify data (UPDATE, DELETE, and INSERT) and for all SQL statements that do not need cursors. Call these numbers C1, C2, ... CN, respectively. For example, if App_1 has three cursors opened concurrently, the value of C1 is 4.
3. Determine the total number of Mode1 sessions needed by location A to access location B by adding $C1+C2+ \dots +CN=LIMIT_A$.
4. If location A also uses Mode1 for the system conversation to location B, then add 1 to LIMIT_A.

5. Repeat steps 1 through 4 for location B applications accessing location A using Mode1. Call this result LIMIT_B.
6. The limit for Mode1 between location A and location B is LIMIT_A+LIMIT_B. This value can be entered as location A's CONVLIMIT for all conversations using Mode1 to access location B.
7. Repeat steps 1 through 6 for every possible mode between location A and location B. Get the maximum of all those numbers, and call it AB_MAX. This result is the maximum session limit between location A and location B.
8. Repeat all the above steps for every possible location that A can connect to.
9. Get the maximum of AB_MAX, AC_MAX, AD_MAX, and so on. This is DSESLIM.

An Example

Assume that Mode2 could possibly be running the three applications described below concurrently.

App_1	App_2	App_3
Has 3 cursors open concurrently referencing location B.	Has 4 cursors open concurrently to location B. Has 6 cursors open concurrently to location C. Mode2 is used for system conversations to location B.	Has no open cursors. References location B only.

You can determine the mode limit for Mode2 as follows:

1. Use the directions in step 2 on page 3-33 to calculate the number of SQL conversations needed for location A to access location B. This number, LIMIT_A, is 10 (4 + 5 + 1).
2. Because Mode2 is used for the system conversation, add 1 to LIMIT_A, which results in a value of 11 for LIMIT_A.
3. Assume that the value of LIMIT_B is 16. Then, the limit for Mode2 as it accesses location B is 27. This value can be entered as the CONVLIMIT value in location A's LUMODES table.
4. The above steps must be repeated for A's connection to location C through App_2. This gives you the maximum number of sessions for Mode2.

This procedure has to be repeated for every mode to determine which mode uses the greatest number of sessions. This, then, is the value of DSESLIM.

Considerations for Mixed Applications

Applications can take advantage of both DB2 private protocol access and DRDA access to data. The following are possible ways to use both DRDA access and DB2 private protocol access in a single application:

- An application at the requesting site accesses a DB2 using DRDA access, then from there, “hops” to another DB2 or DB2s using DB2 private protocol access. This “double hop” application is shown in Figure 106 on page 3-35.

Calculating VTAM I/O Buffer Pool (IOBUF) Storage

This section includes formulas for estimating VTAM buffer pool storage when using DB2's distributed data facility. Every *path information unit* (PIU) that enters or leaves VTAM resides in one or more IOBUF buffers.

A PIU is composed of a 26-byte transmission header, a 3-byte request/response header, and the request/response unit (RU) that contains VTAM application data. You define the length of the RU in a mode entry, using the RUSIZES option.

Do the following to calculate the maximum number of buffers required for the local DB2 subsystem:

1. Calculate the number of buffers that each PIU occupies, and call it PIUBUF.

$$\text{PIUBUF} = (29 + \text{RUSIZE}) / \text{BUFSIZE}, \text{ rounded up to the nearest integer.}$$

RUSIZE is the length of the RU in bytes. It is assumed to be the same for both session directions. BUFSIZE is the value you specified in the IOBUF pool definition.

Assume you have a buffer size of 441 bytes, and an RUSIZE of 4096. With these values, PIUBUF would be 10 $((29+4096) / 441, \text{ rounded up})$.

For channel-to-channel (CTC) and NCP connections, you need to be concerned with the VTAM MAXBFRU value (as shown in "Channel-Connected DB2s" on page 3-42 and "NCP-Connected DB2s" on page 3-43). For CTC connections, MAXBFRU is the number of 4KB buffers allocated to hold the PIUs sent over the channel. If your RU size is 4096 and you allow 29 bytes for the header, then you need to allocate at least 2 4KB buffers. Thus, you need a MAXBFRU value of at least 2.

When you route data through NCP, MAXBFRU is the number of VTAM IOBUF buffers allocated to hold the PIUs sent to the NCP, which means MAXBFRU must be at least as large as PIUBUF.

2. Calculate the maximum number of IOBUF buffers used by a session, and call it SESSBUF.

$$\text{SESSBUF} = \text{PACECNT} \times \text{PIUBUF}$$

PACECNT stands for pacing count. Pacing is discussed in greater detail in "Controlling Pacing" on page 3-26, but for this example we assume that pacing is the same in both directions, and it is the same for all modes.

If pacing is set to 2, then SESSBUF is 20.

3. Calculate the maximum number of sessions that can be active for all modes to all systems and call it SESCNT. Use the formula outlined in "Calculating Session Limits" on page 3-33 to calculate the maximum for each mode, then add those results to get SESCNT.
4. Calculate the maximum number of VTAM buffers used by DB2, and call this DB2BUF. The formula for DB2BUF is based on a worst case scenario, because it assumes that all sessions are used by concurrent conversations.

$$\text{DB2BUF} = \text{SESCNT} \times \text{SESSBUF}$$

If we assume that the maximum number of sessions that can be active is 50 (SESCNT), then 1000 is the number of IOBUF entries required by DB2 in a worst case scenario.

5. Calculate actual VTAM buffer storage consumption used by DB2, and call it STORAGE.

$$\text{STORAGE} = \text{DB2BUF} \times (\text{BUFSIZE} + 71)$$

Each buffer includes 71 bytes for VTAM internal headers.

So, to continue the above example, we can estimate an upper value of real storage as follows:

$$1000 \times (441 + 71) = 500\text{KB}$$

Interpreting CNOS Messages

DB2's distributed data facility can request to alter the number of sessions with another system for a specific VTAM logon mode. This automatic process is called "change number of sessions" (CNOS). This section contains a brief overview of the process as it relates to DB2; it should help you understand the messages that CNOS processing generates. For more information about CNOS processing in general, see *VTAM for MVS/ESA Programming for LU 6.2*

When Sessions Are Started: The AUTOSSES option of the VTAM APPL determines whether, and how many, sessions are started at the time CNOS is negotiated. If AUTOSSES is 0, then the sessions are not started at CNOS negotiation time; they are started as they are needed. We do not recommend an AUTOSSES of 0, because then DB2 is not informed if CNOS fails, and you receive a "resource unavailable" SQL code with the first SQL request to the remote system.

If AUTOSSES is not 0, then sessions are started as follows:

- If AUTOSSES is equal to or less than the number of contention winner sessions for a specific DB2 subsystem, then the number of sessions that are automatically started at CNOS negotiation is equal to AUTOSSES.
- If AUTOSSES is greater than the number of contention winner sessions for a specific DB2 subsystem, only the contention winner sessions are automatically started at CNOS negotiation.

Each LU has its own value for the number of contention winner sessions to start. The total number of sessions started on behalf of a CNOS negotiation request is the sum of the sessions started at each site.

Example: Suppose the DB2 subsystems at USIBMSTODB21 and USIBMSTODB22 have the following values in their VTAM APPL statements and LUMODES tables:


```

*****
**      APPL STATEMENT FOR SYSTEM 1      *
*****
DB1APPL  APPL APPC=YES,                    X
          ATNLOSS=ALL,                     X
          AUTH=(ACQ),                      X
          AUTOSSES=1,                      X
          DSESLIM=20,                      X
          DMINWNL=10,                     X
          DMINWNR=10,                     X
          PRCTCT=DB1PWD,                   X
          SECACPT=ALREADYV,                X
          EAS=509,                         X
          MODETAB=DB2MODES,                X
          PARSESS=YES,                     X
          SRBEXIT=YES,                     X
          SYNCLVL=SYNCPT,                  X
          VPACING=2

*****
**      APPL STATEMENT FOR SYSTEM 2      *
*****
DB2APPL  APPL APPC=YES,                    X
          ATNLOSS=ALL,                     X
          AUTH=(ACQ),                      X
          AUTOSSES=1,                      X
          DSESLIM=20,                      X
          DMINWNL=10,                     X
          DMINWNR=10,                     X
          PRCTCT=DB2PWD,                   X
          SECACPT=ALREADYV,                X
          EAS=509,                         X
          MODETAB=DB2MODES,                X
          PARSESS=YES,                     X
          SRBEXIT=YES,                     X
          SYNCLVL=SYNCPT,                  X
          VPACING=2

*****
**      LOGMODE TABLE FOR SYSTEM 1 TO SYSTEM 2 CONNECTIONS
*****
DB2MODES MODETAB
*****
*      LU6.2 SERVICES MANAGER BASE MODE FOR APPC/VTAM
*****
SNASVCMG MODEENT LOGMODE=SNASVCMG,        X
          FMPPROF=X'13',                   X
          TSPROF=X'07',                   X
          PRIPROT=X'B0',                   X
          SECPROT=X'B0',                   X
          PSERVIC=X'0602000000000000000000300', X
          COMPROT=X'D0B1',                 X
          RUSIZES=X'8585',                 X
          ENCR=B'0000'

*****

```

Figure 111 (Part 1 of 2). Basic VTAM Definitions

```

** DB2 DEFAULT MODE FOR SYSTEM 1 AND SYSTEM 2 SYSTEM-DIRECTED ACCESS**
*****
IBMDB2LM MODEENT LOGMODE=IBMDB2LM, X
                    TYPE=0, X
                    PSNDPAC=X'00', X
                    SSNDPAC=X'02', X
                    SRCVPAC=X'00', X
                    RUSIZES=X'8989', X
                    FMPROF=X'13', LU6.2 FM PROFILE X
                    TSPROF=X'07', LU6.2 TS PROFILE X
                    PRIPROT=X'B0', LU6.2 PRIMARY PROTOCOLS X
                    SECPROT=X'B0', LU6.2 SECONDARY PROTOCOLS X
                    COMPROT=X'50A5', LU6.2 COMMON PROTOCOLS X
                    PSERVIC=X'0602000000000000122F00' LU6.2 LU TYPE
*****
** DB2 DEFAULT FOR SYSTEM 1 AND SYSTEM 2 APPLICATION-DIRECTED ACCESS**
*****
IBMRDB MODEENT LOGMODE=IBMRDB, X
                    TYPE=0, X
                    PSNDPAC=X'00', X
                    SSNDPAC=X'02', X
                    SRCVPAC=X'00', X
                    RUSIZES=X'8989', X
                    FMPROF=X'13', LU6.2 FM PROFILE X
                    TSPROF=X'07', LU6.2 TS PROFILE X
                    PRIPROT=X'B0', LU6.2 PRIMARY PROTOCOLS X
                    SECPROT=X'B0', LU6.2 SECONDARY PROTOCOLS X
                    COMPROT=X'50A5', LU6.2 COMMON PROTOCOLS X
                    PSERVIC=X'0602000000000000122F00' LU6.2 LU TYPE
*****
** DB2 SYSTOSYS MODE FOR SYSTEM 1 AND SYSTEM 2 *
*****
SYSTOSYS MODEENT LOGMODE=SYSTOSYS, X
                    TYPE=0, X
                    PSNDPAC=X'00', X
                    SSNDPAC=X'02', X
                    SRCVPAC=X'00', X
                    RUSIZES=X'8989', X
                    FMPROF=X'13', LU6.2 FM PROFILE X
                    TSPROF=X'07', LU6.2 TS PROFILE X
                    PRIPROT=X'B0', LU6.2 PRIMARY PROTOCOLS X
                    SECPROT=X'B0', LU6.2 SECONDARY PROTOCOLS X
                    COMPROT=X'50A5', LU6.2 COMMON PROTOCOLS X
                    PSERVIC=X'0602000000000000122F00' LU6.2 LU TYPE
                    MODEEND END
*****
** ATCSTRTA VTAM START OPTIONS FOR SYSTEM 1, INCLUDES IOBUF
*****
CONFIG=TA, X
SSCPID=53,MAXSUBA=150,HOSTSA=53, X
SSCPNAME=SSCP004,NETID=USIBMSY, X
IOBUF=(328,441,20,,64,48,768)
*****
** ATCSTRTB VTAM START OPTIONS FOR SYSTEM 2, INCLUDES IOBUF
*****
CONFIG=TB, X
SSCPID=54,MAXSUBA=150,HOSTSA=54, X
SSCPNAME=SSCP00E,NETID=USIBMSY, X
IOBUF=(328,441,20,,64,48,768)

```

Figure 111 (Part 2 of 2). Basic VTAM Definitions

Channel-Connected DB2s

When determining your channel-to-channel definitions, remember that MAXBFRU must be large enough to handle the largest PIU. Since DB2 is sending 4096 bytes, you need enough 4KB buffers to accept 4096 + 29 bytes (the 29 bytes is for the network header). Thus MAXBFRU must be at least 2 in our example.

In many cases, the DB2 RU size is larger than any other PIUs used on existing CTCs, which can mean you must examine your MAXBFRU values on existing CTC definitions. If the values are too small, you get an SNA X'800A' sense code, indicating that the PIU was truncated during transmission.

```
*****
**          CTC DEFINITIONS FOR SYSTEM 1          *
*****
DB1CTC  VBUILD TYPE=CA          CTC MAJOR NODE DEFINITION
DB1GRPB GROUP LNCTL=CTCA,      CTCA LINE TYPE                X
          MIH=YES,REPLYTO=10.0
DB1CTCL LINE ADDRESS=(500),    CTC ADDRESS FOR THIS LINE    X
          DELAY=0,             CTC DELAY                    X
          MAXBFRU=8,          MAX BUFFER USED                X
          ISTATUS=ACTIVE      INITIAL STATUS IS ACTIVE
DB1CTCP PU  ISTATUS=ACTIVE

*****
**          CTC DEFINITIONS FOR SYSTEM 2          *
*****
DB2CTC  VBUILD TYPE=CA          CTC MAJOR NODE DEFINITION
DB2GRPB GROUP LNCTL=CTCA,      CTCA LINE TYPE                X
          MIH=YES,REPLYTO=10.0
DB2CTCL LINE ADDRESS=(500),    CTC ADDRESS FOR THIS LINE    X
          DELAY=0,             CTC DELAY                    X
          MAXBFRU=8,          MAX BUFFER USED                X
          ISTATUS=ACTIVE      INITIAL STATUS IS ACTIVE
DB2CTCP PU  ISTATUS=ACTIVE

*****
**          PATH - NETWORK ROUTES FOR SYSTEM 1   *
*****
MVSDB2  PATH DESTSA=2,ER1=(2,1),VR1=1,          X
          VRPWS10=(2,30),VRPWS11=(2,30),VRPWS12=(2,30)
*****
**          PATH - NETWORK ROUTES FOR SYSTEM 2   *
*****
MVSDB1  PATH DESTSA=1,ER1=(1,1),VR1=1,          X
          VRPWS10=(2,30),VRPWS11=(2,30),VRPWS12=(2,30)
*****
**          CDRSC DEFINITIONS FOR SYSTEM 1      *
*****
          VBUILD TYPE=CDRSC
DB2APPL CDRSC CDRM=DB2CDRM, ISTATUS=ACTIVE
*****
**          CDRSC DEFINITIONS FOR SYSTEM 2      *
*****
          VBUILD TYPE=CDRSC
DB1APPL CDRSC CDRM=DB1CDRM, ISTATUS=ACTIVE
```

Figure 112 (Part 1 of 2). Channel-to-Channel (CTC) Definitions


```

*****
*          PCCU SPECIFICATION - FOR SYSTEM 1          *
*****
PCCU1  PCCU  CUADDR=C02,          3745 BLOCK CHANNEL          X
          AUTOSYN=YES,          X
          AUTODMP=NO,          X
          AUTOIPL=NO,          X
          BACKUP=YES,          X
          DELAY=0,          X
          DUMPDS=DUMPDS,        DUMP DATA SET          X
          CDUMPDS=CDUMPDS,     CSP DUMP DATA SET       X
          MDUMPDS=MDUMPDS,     MOSS DUMP DATA SET      X
          INITEST=NO,          NO 3745 INITIAL TESTS AT LOAD TIME X
          MAXDATA=4302,        = BFRS*TRANSFR - 18      X
          OWNER=HOST1,          X
          SUBAREA=3,           HOST SUBAREA              X
          VFYLM=YES
*****
*          PCCU SPECIFICATION - FOR SYSTEM 2          *
*****
PCCU2  PCCU  CUADDR=C02,          3745 BLOCK CHANNEL          X
          AUTOSYN=YES,          X
          AUTODMP=NO,          X
          AUTOIPL=NO,          X
          BACKUP=YES,          X
          DELAY=0,          X
          DUMPDS=DUMPDS,        DUMP DATA SET          X
          CDUMPDS=CDUMPDS,     CSP DUMP DATA SET       X
          MDUMPDS=MDUMPDS,     MOSS DUMP DATA SET      X
          INITEST=NO,          NO 3745 INITIAL TESTS AT LOAD TIME X
          MAXDATA=4302,        = BFRS*TRANSFR - 18      X
          OWNER=HOST2,          X
          SUBAREA=4,           HOST SUBAREA              X
          VFYLM=YES

```

Figure 113 (Part 1 of 4). Network Control Program (NCP) Definitions

```

*****
*          BUILD MACRO SPECIFICATIONS          *
*****
ACFNCPBD BUILD BFRS=(240),          NCP BUFFER SIZE,# EP BUFFERS      X
                BRANCH=1000,                X
                CATRACE=(YES,10),            X
                CSMHDR=27F5C711C3F0405C40C8C4D9405C,          X
                CSMHDRC=40E3C5E7E3405C5C,          X
                CSMSG=5C5C40E5E3C1D440E2C8E4E3C4D6E6D540,      X
                CSMSGC=6040C8C1E240C2C5C7E4D5405C5C,          X
                DIALTO=60,                    WAIT 1 MIN FOR AUTOCALL ANSWER  X
                DR3270=NO,                    NO DYNAMIC RECONFIG          X
                DSABLT0=3.0,                  TIME TO DETECT DSR DROP      X
                ENABLT0=2.2,                  TIME TO DETECT DSR AFTER ENABLE X
                LOADLIB=NCPLoad,              LIBRARY FOR ACF/NCP LOAD MODULE X
                LTRACE=8,                     UP TO 8 LINES CONCURRENTLY TRACED X
                MAXSSCP=8,                    NUMBER OF SSCPS IN SESSION   X
                MAXSUBA=63,                   MUST BE SAME AS IN ATCSTRXX  X
                MEMSIZE=4M,                   AMOUNT OF MEMORY            X
                MODEL=3745,                   3745 MODEL 410              X
                NETID=BCR1,                   3745 MODEL 410              X
                NEWNAME=DDBLCO,              LOAD MODULE NAME            X
                PUNAME=DDB,                  X                             X
                NPA=YES,                      NPA WILL NOT COLLECT DATA   X
                OLT=NO,                       INCLUDE ONLINE TEST FACILITY-OLTEP X
                PRTGEN=NOGEN,                 DON'T PRINT ASSEMBLED STATEMENTS X
                PWROFF=NO,                    X                             X
                SLOWDOWN=15,                  SLOWDOWN AFTER 15% BUFFERS AVAIL X
                SUBAREA=26,                   NCP SUBAREA                  X
                TRACE=(YES,10),               10-16 BYTE ADDRESS TRACE ENTRIES X
                TRANSFR=18,                   =(4096+51)/BFRS--ROUNDED UP   X
                TRCPIU=2000,                  SIZE OF LINE AND SIT TRACE    X
                TYPGEN=NCP,                   X                             X
                TYP SYS=MVS,                   MVS OPERATING SYSTEM         X
                USGTIER=5,                     NCP USAGE TIER - REQUIRED      X
                VERSION=V5R3,                  X                             X
                XBREAK=NONE

```

Figure 113 (Part 2 of 4). Network Control Program (NCP) Definitions

```

*****
**                                           **
*          SYSCNTRL OPTIONS - REQUIRED BY VTAM          *
**                                           **
*****
                SYSCNTRL OPTIONS=(ENDCALL,MODE,RCNTRL,RCOND,RECMD,RIMM,          X
                SESSION,NAKLIM,LNSTAT,SSPAUSE,XMTLMT,BHSASSC,STORDSP)
*****
HOST2  HOST  BFRPAD=0,          VTAM REQUIREMENT FOR OS          X
                INBFRS=18,        INITIAL BUFFERS FOR EACH RECEIVE X
                MAXBFRU=10,        < BASENO IN IOBUF FOR VTAM    X
                SUBAREA=4,         X                             X
                UNITSZ=441         = BUFSIZE IN IOBUF FOR VTAM
HOST1  HOST  BFRPAD=0,          VTAM REQUIREMENT FOR OS          X
                INBFRS=18,        INITIAL BUFFERS FOR EACH RECEIVE X
                MAXBFRU=10,        < BASENO IN IOBUF FOR VTAM    X
                SUBAREA=3,         X                             X
                UNITSZ=441         = BUFSIZE IN IOBUF FOR VTAM
*****
*          PATH STATEMENTS          *
*****
                PATH DESTSA=3,        SYS1          X
                ER4=(3,1),            SYS1
                PATH DESTSA=4,        SYS2          X
                ER4=(4,1),            SYS2

```

Figure 113 (Part 3 of 4). Network Control Program (NCP) Definitions

```

*****
*
*   HOST 1 CHANNEL ADAPTER
*
*   LINE ADDR = 0; PHYSICAL POSITION = 5.
*****
DDBCA5  GROUP LNCTL=CA,                                X
        ISTATUS=INACTIVE   STOP VTAM FROM ACT CHAN LINK

DDBL05  LINE ADDRESS=0,          1ST CA PHYSICAL POSITION 1      X
        CA=TYPE6,           3745 CHANNEL ADAPTER TYPE          X
        CASDL=120,          TIME ALLOWED TO BLOCK INBOUND DATA X
        DELAY=0,            CHAN ATTN DELAY                     X
        DYNADMP=NONE,       NO EP SUBCHANNELS TO DUMP           X
        INBFRS=18,          # BUFS FOR EACH TRANSFER TO HOST     X
        NPACOLL=YES,        NPA WILL COLLECT DATA ON CHANNEL   X
        TIMEOUT=120         INTERVAL BEFORE CHANNEL DISCONTACT

DDBP05  PU   PUTYPE=5,          INTERMEDIATE SUBAREA FUNCTION   X
        TGN=1                 MUST BE 1 FOR PUTYPE5

*****
*
*   HOST 2 CHANNEL ADAPTER
*
*   LINE ADDR = 2; PHYSICAL POSITION = 7.
*****
DDBCA7  GROUP LNCTL=CA,                                X
        STATUS=INACTIVE   ACT CHAN LINK

DDBL07  LINE ADDRESS=2,          3RD CA PHYSICAL POSITION 3      X
        CA=TYPE6,           3745 CHANNEL ADAPTER TYPE          X
        CASDL=120,          TIME ALLOWED TO BLOCK INBOUND DATA X
        DELAY=0,            CHAN ATTN DELAY                     X
        DYNADMP=NONE,       NO EP SUBCHANNELS TO DUMP           X
        INBFRS=18,          #BUFS FOR EACH TRANSFER TO HOST     X
        NPACOLL=YES,        NPA WILL COLLECT DATA ON CHANNEL   X
        TIMEOUT=120         INTERVAL BEFORE CHANNEL DISCONTACT

DDBP07  PU   PUTYPE=5,          INTERMEDIATE SUBAREA FUNCTION   X
        TGN=1                 MUST BE 1 FOR PUTYPE5

```

Figure 113 (Part 4 of 4). Network Control Program (NCP) Definitions

Using the Change Log Inventory Utility to Update the BSDS

Use the options of the DDF statement of the change log inventory utility to insert or update the values listed below.

To **update** any value, you need only the option for that value.

To **insert** new values, you need values for LOCATION and LUNAME as well as any other values.

Value	Option for inserting or updating
Location name	LOCATION= <i>name</i>
LU name	LUNAME= <i>name</i>
Generic LU name	GENERIC= <i>name</i>
Password	PASSWORD= <i>password</i>

PASSWORD is optional, depending on whether you entered a password in the VTAM APPL statement. GENERIC is also optional.

To **delete** either a generic LU name or a password, use one of these statements:

Value	Statement for deleting
Generic LU name	DDF NGENERIC
Password	DDF NOPASSWD

For more information about the change log inventory utility, see Section 3 of *Utility Guide and Reference*.

Chapter 3-3. Connecting Systems with TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) is a standard communication protocol for network communications. Previous versions of DB2 supported TCP/IP requesters, although additional software and configuration was required. Native TCP/IP eliminates these requirements, allowing gateway-less connectivity to DB2 for systems running MVS OpenEdition.

Terminology: The following communications terms are used in this chapter:

IP address

Uniquely identifies a host within the TCP/IP network. This is sometimes called an internet address. A DB2 subsystem resides on a TCP/IP host. The IP address is a four-byte address displayed in dotted decimal format: X'05041020' displays as 5.4.16.32

Domain name

The fully qualified name that identifies an IP address. This can be used instead of the IP address. An example of a domain name is *stlmvs1.stl.ibm.com*. Some software refers to *stlmvs1* as the host name and *stl.ibm.com* as the domain name. DB2 allows the network administrator to identify a host using a domain name.

Domain name server (DNS)

Manages a distributed directory of domain names and related IP addresses. Domain names can be translated into IP addresses and you can find a domain name associated with a given IP address. DB2 uses the *gethostbyname* service to get a list of IP addresses for a given domain name.

Port

Identifies an application executing in a host. For example, a port number identifies a DB2 subsystem to TCP/IP. A port number is a two byte integer value that is displayed in decimal format. This number identifies the application within a TCP/IP instance. A port number of X'01D2' displays as 466. There are three basic kinds of TCP/IP ports:

Well-known port

This is a port number between 1 and 1023 that is reserved in the TCP/IP architecture for a specific TCP/IP application. Some typical well-known port numbers are:

- FTP is port number 21
- Telnet is port number 23
- DRDA relational database is port number 446

Ephemeral port

Port numbers that are dynamically assigned to a client process by the client's TCP/IP instance. DB2 uses an ephemeral port when it is acting as the DRDA application requestor (AR). This ephemeral port is associated with the requester for the life of the thread, or connection.

Server port

Port numbers that are used when a TCP/IP program does not have a well-known port number, or another instance of the server program is already installed using the well-known port number. As a requester, DB2 defaults to using the DRDA relational database well-known port number to connect to a

server location. We suggest that your DB2 subsystem be defined with the DRDA well-known port number of 446. However, the network administrator can assign a server port to the DB2 subsystem.

If two different DB2 subsystems reside on the same host, acting as two different locations (a non-data-sharing group), each DB2 subsystem must have a unique port. In this case, only one DB2 subsystem can use the DRDA well-known port number.

Service name

Another way to refer to a port number. A network administrator can assign a service name for a remote location instead of using the port number.

The domain name (IP address) and service name (port number) uniquely identify a DB2 subsystem in the TCP/IP network. The domain name and the service name of the database server **must** be defined in the communications database (CDB) so that a DB2 subsystem can connect to a remote location. The domain name and service name **must** be defined to the TCP/IP host so that a DB2 subsystem can accept connections from remote locations. When DDF is started, the DB2 subsystem binds itself to the port.

Enabling TCP/IP communication

These steps enable TCP/IP communication between DRDA partners and DB2. You do not have to do the steps in any particular order, but steps 1, 2, and 3 should be completed prior to the other steps.

- “Step 1: Install OpenEdition MVS and TCP/IP and Prepare the LE/370 Runtime Library” on page 3-52.
- “Step 2: Enable DDF for OpenEdition” on page 3-52.
- “Step 3: Define the DB2 Subsystem to TCP/IP” on page 3-53.
- “Step 4: Populate the Communications Database” on page 3-56.
- “Step 5: Start TCP/IP support” on page 3-58.
- “Step 6: Tuning TCP/IP” on page 3-59.

If you do not use VTAM to communicate with remote sites, you still must define VTAM to DB2 as described in “Chapter 3-2. Connecting Systems with VTAM” on page 3-9 because DB2 TCP/IP communications uses NETID and LUNAME to identify units of work.

See Chapter 5 of *Data Sharing: Planning and Administration* for information about TCP/IP and data sharing.

DDF enhancements enable TCP/IP communication with DRDA partners with DRDA Level 3 TCP/IP support. OS/2 and AIX clients must have the updated versions of DB2 Connect or any DRDA requester or server that supports DRDA Level 3. This DRDA TCP/IP connectivity lets you connect DDF to clients on multiple platforms directly, without an intermediate DRDA LAN server (DB2 Connect enterprise server). Remember, to connect without the SNA gateway, the client needs the DRDA (DB2 Connect) support installed. If DRDA is not available on the client, you will need to connect using a DRDA LAN server (DB2 Connect enterprise server).

You can have DB2 send or receive requests using SNA, TCP/IP, or both network protocols. For more information, see “How the Communications Database (CDB) Handles Requests” on page 3-5.

TCP/IP Limitations

MVS OpenEdition supports only one TCP/IP instance, so you cannot have multiple TCP/IP instances on the same system.

TCP/IP does not support DB2 private protocol connections, but you can use SNA for DB2 private protocol connections while using TCP/IP for DRDA connections.

TCP/IP does not have built-in security features that SNA has, such as SNA partner LU verification. Because IP addresses are not as reliable as LU names, DDF support for TCP/IP differs from support for SNA in these ways:

- There is no support for inbound name translation.
- A DB2 system parameter, TCPALVER in DSN6FAC, defines the minimum security requirements for all TCP/IP clients because inbound security requirements cannot be established on individual clients.
- You cannot use the CDB for *come from* checking of TCP/IP clients.

Using Two-phase Commit

DB2 supports two types of 2-phase commit for TCP/IP clients:

- The DRDA client coordinates the 2-phase commit. If a failure occurs during the commit process, DB2 might need to resynchronize with the DRDA client.
- The DRDA client gives responsibility for the resynchronization to DB2. The client sends DB2 a list of server LOCATION names and IP addresses that are part of the client's unit of work. If a failure occurs during the 2-phase commit process, DB2 might need to resynchronize with one or more of the server locations sent by the client.

DB2 uses the port specified on the RESYNC PORT field of installation panel DSNTIP5 for 2-phase commit resynchronization. DB2 begins resynchronization using the partner's IP address and LOCATION name, and the RESYNC PORT obtained at the time of initial connection. If the partner's IP address changed, the resynchronization fails. For example, if the partner was DB2 for OS/390, the IP address can change when the automatic restart manager (ARM) restarts a data sharing member on a different CPC. The IP address can also change when an MVS adapter fails and virtual IP addresses were not used.

If the IP address fails, DB2 uses the partner's domain name to determine the IP address for resynchronization. DRDA requesters receive the port number and domain name to be used for 2-phase commit resynchronization from the server during DRDA connect processing.

No CDB definition is required to do resynchronization.

Step 1: Install OpenEdition MVS and TCP/IP and Prepare the LE/370 Runtime Library

Make sure that OpenEdition MVS sockets and TCP/IP are both installed and
configured . Please refer to *IBM TCP/IP for MVS: Planning and Migration Guide* for
| installation information for TCP/IP, and *OS/390 OpenEdition Planning* for installation
| information for OpenEdition MVS.

Because DDF uses some functions in the LE/370 library, DDF needs access to the runtime library. The standard way to handle this is to include the LE/370 library in a STEPLIB concatenation for the DDF JCL procedure. The LE/370 library must be APF authorized to be added to the DDF JCL procedure. The DB2 installation automatically adds the library to the DDF STEPLIB concatenation.

The alternative method is to concatenate the LE/370 library in the MVS link list, which does not require the library to be APF authorized. If you choose this alternative method, remove the LE/370 library concatenation from the DDF JCL procedure.

Step 2: Enable DDF for OpenEdition

DDF uses the OpenEdition asynchronous I/O assembler callable interface to perform TCP/IP services. These functions require DDF to execute as an authorized user with the appropriate privileges. DDF must execute in the OpenEdition environment as a superuser to perform asynchronous I/O socket calls. If the socket call fails because of insufficient authorization (OpenEdition reason code=1148033C) refer to message DSNL512 in *Messages and Codes*.

Please note that DDF executes as an authorized program and is protected against any unauthorized use of this privilege by DDF users.

To enable DDF as a superuser, the RACF USERID for the DDF started task *ssnmDIST* address space is required to have a USERID of zero.

To enable DDF as a superuser, issue one of the following RACF commands:

```
ADDUSER ddfuid OMVS(UID(0))...  
ALTUSER ddfuid OMVS(UID(0))...
```

Replace *ddfuid* in the above commands with the MVS user ID associated with the DDF address space.

If you specify both a userid and a group in the RACF Started Procedures table, ICHRIN03, then the group must have a valid OpenEdition group ID setting. You define RACF groups to be OpenEdition groups with one of the following commands:

```
ADDGROUP ddfuid OMVS(GID(x))...  
ALTGROUP ddfuid OMVS(GID(x))...
```

where *x* is any valid, unique identifier.

Replace *ddfuid* in the above commands with the MVS group ID associated with the
DDF address space.

These associations (UID and GID) are made through the RACF Started Procedures table, ICHRIN03. For more information on this table, please refer to *Recourse Access Control Facility (RACF) System Programmer's Guide*.

Step 3: Define the DB2 Subsystem to TCP/IP

The DB2 subsystem uses different TCP/IP ports to do different tasks:

- As a requester, DB2 uses an ephemeral port. You do not need to specify this port.
- As a server processing TCP/IP connection requests for DRDA SQL applications, DB2 uses a server port or the well-known port, 446, which is used for relational database communications.
- A server resynchronization port is used for processing 2-phase commit resynchronization requests.

This requires some planning because the port number is used to pass the network requests to the right DB2 subsystem. Each location must have a unique port number. *Data Sharing: Planning and Administration* has information for assigning port numbers for systems that have enabled data sharing.

Figure 114 on page 3-54 shows some typical MVS system configurations that demonstrate some typical configurations.

- In SYSTEM1, there is only one DB2 subsystem, so the DRDA well-known port (446) can be assigned to DB2. In the example, port number 5020 is assigned for 2-phase commit resynchronization.
- In SYSTEM2, there are two DB2 subsystems, making it impossible to assign the port numbers 446 and 5020 to both DB2 subsystems, because TCP/IP can only support one server at each port number. The problem is resolved by assigning the 446 and 5020 port numbers to DB2C, and port numbers 5021 and 5022 to DB2D.

Be sure to consider the impact of future system consolidations. If SYSTEM1 and SYSTEM2 are consolidated so that DB2A, DB2C, and DB2D run on a single MVS system, you must take special precautions because DB2A and DB2C have the same TCP/IP port numbers. You can resolve this by changing the port numbers of either DB2A or DB2C to eliminate the duplicate port numbers.

3. Define the TCP/IP host names that DB2 needs to know.

The local host name **must** be defined before DDF is started. All domain names referenced in the table SYSIBM.IPNAMES **must** be defined. You define the host names by configuring the *hlq*.HOSTS.LOCAL data set, the */etc/hosts* file in the hierarchical file system (HFS), or the domain name server (DNS).

After the *hlq*.HOSTS.LOCAL data set is configured, you have to execute the utility MAKESITE. This utility generates the *hlq*.HOSTS.ADDRINFO and the *hlq*.HOSTS.SITEINFO data sets that are used to translate between domain names and IP addresses. MAKESITE generates under the userid that issued the MAKESITE. Therefore, those files have to be moved to the high level qualifier that represents TCP/IP. The host name for the local DB2 subsystem must be defined in at least one of these places.

If domain names are present in the CDB (in field IPADDR of table SYSIBM.IPNAMES), they must be defined in the MVS data sets, the HFS or the DNS.

To support a Sysplex environment or multiple network interfaces on a single host, we recommend that the DNS or *hlq*.HOSTS.LOCAL data set be used to associate multiple IP addresses with a single host name.

The following example from Figure 114 on page 3-54 shows a sample *hlq*.HOSTS.LOCAL entry for SYSTEM2:

```
HOST : 121.65.183.98 : S2.VNET.IBM.COM
```

4. Define the TCP/IP service names that DB2 needs to know.

Configure the *hlq*.ETC.SERVICES data set or the */etc/services* file in the HFS. If service names are present in the CDB (in field PORT of table SYSIBM.LOCATIONS), they must be defined in the MVS data set or the HFS.

The following example shows a sample *hlq*.ETC.SERVICES entry:

```
DRDA          446/tcp          ; DRDA databases
```

For more detailed information on these steps see *IBM TCP/IP for MVS: Customization & Administration Guide*.

Modify the Change Log Inventory Job

To use TCP/IP, the DDF statement of the change log inventory job (DSNJU003) must specify values for the parameters PORT and RESPORT.

The parameter PORT is the TCP/IP port number used by DDF to accept incoming DRDA connection requests. The parameter RESPORT is the TCP/IP port number used by DDF to accept incoming DRDA 2-phase commit resynchronization requests. The values for each of these parameters must be a decimal number between 0 and 65534, where zero indicates that DDF's TCP/IP support is being deactivated. The non-zero value for PORT must not be the same as the non-zero value for RESPORT.

For data sharing, all the members of the DB2 data sharing group must have the same value for PORT. RESPORT must be uniquely assigned to each DB2 member so that no two DB2 members use the same TCP/IP port for 2-phase commit resynchronization. The parameters PORT and RESPORT can be changed on any DB2 member by running the utility change log inventory. After running the utility,

you must stop and then restart DDF. Since PORT is the same for all members of the DB2 group, this process has to be repeated on every member of the group when PORT is changed.

Remember, a zero value for either PORT or RESPORT is the same as deactivating DB2's TCP/IP support.

Step 4: Populate the Communications Database

The information under this heading, up to “Step 5: Start TCP/IP support” on page 3-58 is General-use Programming Interface, as defined in “Notices” on page ix.

If you plan to use DB2 only as a server, you do not need to populate the CDB. For example, Spiffy's USIBMSTODB21 subsystem works as a server for many OS/2 requesters. It is not necessary for Spiffy to register those requesters in DB2's CDB.

However, if you intend to request data, you need to enter port numbers or service names in field PORT of table SYSIBM.LOCATIONS, and IP addresses or domain names in field IPADDR of table SYSIBM.IPNAMES. The LINKNAME in table SYSIBM.LOCATIONS is used to search tables SYSIBM.IPNAMES and SYSIBM.LUNAMES (see “Chapter 3-2. Connecting Systems with VTAM” on page 3-9). If RACF PassTickets are used, the LINKNAME must match the LUNAME of the remote site. Section 3 (Volume 1) of *Administration Guide* discusses the requirements for the other tables.

After you populate these tables, you can write queries that access data at a remote system. For instructions on sending SQL statements to other systems, see *Application Programming and SQL Guide*. For instructions on granting privileges to users on remote DB2 subsystems, see Section 3 (Volume 1) of *Administration Guide*.

SYSIBM.LOCATIONS Table

The table LOCATIONS is used to determine the port number or service name used to connect to the remote location. The column LINKNAME maps to the corresponding row in table IPNAMES.

LOCATIONS has the following columns relating to TCP/IP:

LOCATION CHAR(16)

The unique network location name, or DRDA RDBNAM, assigned to a system, remote or local. You must provide location names for any systems that you request data from. This column is the primary key for this table.

LINKNAME CHAR(8)

Identifies the TCP/IP attributes associated with this location. For each LINKNAME specified, you must have a row in SYSIBM.IPNAMES whose LINKNAME matches the value specified in this column. Because this table is used for outbound requests, you must provide a LINKNAME or your requests fail. Do not enter blanks in this column.

PORT CHAR(32)

If blank, the default port, 446, is used for TCP/IP communications. Otherwise, the value can be either:

- The port number of the remote database server. The number must be 1-5 characters and left justified.
- A TCP/IP service name. The service name is converted to a TCP/IP port number with the *getservbyname* socket call.

Spiffy's USIBMSTODB21 location wants a LOCATIONS table that looks like Table 126. The location USIBMSTODB21 uses the default DRDA PORT, 446.

Table 126. Spiffy's LOCATIONS Table

LOCATION	LINKNAME	PORT
USIBMSTODB21	LUDB21	
USIBMSTODB22	LUDB22	
USIBMSTOSQL1	LUSQLDS	1234
USIBMSTOSQL2	LUSQLDS	DRDA

For example, add the second row with this statement:

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME)
VALUES ('USIBMSTODB22', 'LUDB22');
```

Since no port number is specified, location USIBMSTODB22 uses the default DRDA port number, 446.

A Row for the Local Location: You do not need a row for the local DB2 in the IPNAMES and LOCATIONS tables. For example, Spiffy's USIBMSTODB21 subsystem does not require a row that shows its own LINKNAME and location name.

SYSIBM.IPNAMES Table

IPNAMES defines the outbound security and host names used to connect to other systems using TCP/IP. IPNAMES has the following columns:

LINKNAME CHAR(8)

This value matches that specified in the LINKNAME column of the associated row in SYSIBM.LOCATIONS.

SECURITY_OUT CHAR(1)

Defines the security option that is used when local DB2 SQL applications connect to any remote server associated with this TCP/IP host. The default, A, means that outgoing connection requests contain an authorization ID without a password.

USERNAMES CHAR(1)

This column is used for outbound requests to control translations of authorization IDs. The values 'O' or 'B' are valid for TCP/IP connections.

IPADDR VARCHAR(254)

This column contains the IP address or domain name of a remote TCP/IP host.

- An IP address must be 1 to 15 characters and left justified. An example of an IP address is *9.112.46.111*.
- A domain name is converted to an IP address by the domain name server. An example of a domain name is

stlmvs1.stl.ibm.com. The *gethostbyname* socket call is used to resolve the domain name.

SYSIBM.USERNAMES Table

USERNAMES contains information needed for outbound translation only.

Reminder: Inbound ID translation and *come from* checking are not done for TCP/IP requesters.

TYPE CHAR(1)

Whether the row is for outbound translation. The value 'O' is valid for TCP/IP connections.

AUTHID CHAR(8)

Authorization ID to translate. If blank, it applies to all authorization IDs.

LINKNAME CHAR(8)

Identifies the TCP/IP network location associated with the row. A blank indicates it applies to all TCP/IP partners. For nonblank values, this value must match the LINKNAME value in SYSIBM.IPNAMES.

NEWAUTHID CHAR(8)

The translated value of AUTHID.

PASSWORD CHAR(8)

The password to accompany an outbound request. This column is ignored if RACF PassTickets, or already verified USERIDs are used.

Step 5: Start TCP/IP support

Before you start DDF, OpenEdition MVS must be started in either "full function" or
"sockets-only" mode, TCP/IP must be started and the local host name must be
defined. The local host name is defined in */etc/hosts*, *TCPIP.HOSTS.LOCAL*, or the
domain name server (DNS).

DDF executes the following steps when it is started:

1. Notify OpenEdition MVS that DDF requires the asynchronous I/O TCP/IP stack.
2. Obtain the local host information from OpenEdition MVS. Issue the TCP/IP *gethostid* socket call to get the local IP address. Issue the TCP/IP *gethostbyaddr* socket call to get the local domain name.

Note: Until the local host information is available, DDF TCP/IP services are not available to local and remote applications. If a failure occurs obtaining the local host information, DDF periodically attempts to get the local host information until successful, or until DDF is stopped.

3. Set the maximum number of pending connection requests on a TCP/IP socket.
4. Establish a TCP/IP socket for the DRDA and resync port used to accept connections from remote locations.

The following socket options are used when establishing a TCP/IP socket:

SO_LINGER

Indicates to linger on a close connection if data is present.

SO_REUSEADDR

Allows local addresses to be reused.

SO_KEEPALIVE

Used to test if a dormant connection is still alive.

Step 6: Tuning TCP/IP

This is an optional step but the recommendations here can protect DB2 from TCP/IP outages.

The recommendations are:

- Use the IDLE THREAD TIMEOUT installation option on the Distributed Data Facility panel, DSNTIPR, to limit the time an idle thread can hold locks.
- Specify a small value, 5 minutes or less, for the TCP/IP keep_alive timer. If the network fails between the server's reply and the next client request, TCP/IP waits until the keep_alive timer expires, then notifies the DB2 subsystem of the failure.

The server thread hangs while the timer is running. Because the timer default is 2 hours, threads can hang for up to 2 hours if you use the default. The hung thread can cause unpredictable results, depending on what resources it has locked.

If you are connecting to a location whose LINKNAME is associated with a row in the table SYSIBM.IPNAMES and it has a domain name in the IPADDR field, *gethostbyname* can return a list of IP addresses associated with the LINKNAME. When trying to connect to this location, DB2 will try each of these IP addresses in a round-robin fashion (starting with the first address) until the connection is successful, or the attempts to connect to each IP address has timed out.

Appendixes

Appendix A. Sample Application Output	X-3
Phase 1 (DSNTEJ1P Job)	X-3
Phase 2	X-11
DSNTEJ2A Job	X-11
DSNTEJ2C Job	X-16
DSNTEJ2D Job	X-19
DSNTEJ2E Job	X-22
DSNTEJ2F Job	X-25
DSNTEJ2P Job	X-28
Phase 3 (DSNTEJ3P Job)	X-31
Phases 4 and 5	X-53
Appendix B. Character Conversion	X-55
Understanding Character Conversion	X-55
Specifying a System Coded Character Set Identifier	X-56
Special considerations	X-59
Converting to the Euro symbol	X-60
Customizing Character Conversion	X-62
How an Entry in SYSIBM.SYSSTRINGS Works	X-62
Adding a Character Conversion Procedure to DB2	X-63
Rules for SYSSTRINGS Entries	X-64
When Remote Packages Should be Rebound	X-64

#

Appendix A. Sample Application Output

The information in this appendix is General-use Programming Interface and Associated Guidance Information as defined in “Notices” on page ix.

This appendix contains information useful for evaluating the output you get when you run the sample applications during installation verification. It contains parts of the sample output from the jobs in Phases 1, 2, and 3. For all other jobs, see the return code tables in “Chapter 2-9. Verifying with the Sample Applications” on page 2-305.

Because of differences in system configurations, your sample output may differ from the output shown here.

Phase 1 (DSNTEJ1P Job)

Here is part of the sample output for DSNTEJ1P:

```
READY
DSN SYSTEM(V51A)
DSN
BIND PACKAGE (DSNTEP2) MEMBER(DSNTEP2) ACT(REP) ISO(CS)
WARNING, ONLY IBM-SUPPLIED COLLECTION-IDS SHOULD BEGIN WITH "DSN"
WARNING, ONLY IBM-SUPPLIED PACKAGE-IDS SHOULD BEGIN WITH "DSN"
DSNT254I . DSNTEP2 BIND OPTIONS FOR
      PACKAGE = SANTA_TERESA_LAB.DSNTEP2.DSNTEP2. ()
      ACTION      REPLACE
      OWNER       SYSADM
      QUALIFIER   SYSADM
      VALIDATE    RUN
      EXPLAIN     NO
      ISOLATION   CS
      RELEASE
      COPY
DSNT255I . DSNTEP2 BIND OPTIONS FOR
      PACKAGE = SANTA_TERESA_LAB.DSNTEP2.DSNTEP2. ()
      SQLERROR    NOPACKAGE
      CURRENTDATA YES
      DEGREE      1
      DYNAMICRULES
      DEFER
      NOREOPT     VARS
      KEEPYNAMIC  NO
DSNT232I . SUCCESSFUL BIND FOR
      PACKAGE = SANTA_TERESA_LAB.DSNTEP2.DSNTEP2. ()
```

Figure 115 (Part 1 of 9). Sample Output from Job DSNTEJ1P

```

DSN
  BIND PLAN(DSNTEP51) PKLIST(DSNTEP2.*) ACT(REP) ISO(CS) SQLRULES(DB2)
WARNING, ONLY IBM-SUPPLIED PLAN-NAMES SHOULD BEGIN WITH "DSN"
DSNT252I . DSNTBCM1 BIND OPTIONS FOR PLAN DSNTEP51
      ACTION          REPLACE
      OWNER           SYSADM
      VALIDATE        RUN
      ISOLATION       CS
      ACQUIRE        USE
      RELEASE         COMMIT
      EXPLAIN         NO
      DYNAMICRULES   RUN
DSNT253I . DSNTBCM1 BIND OPTIONS FOR PLAN DSNTEP51
      NODEFER        PREPARE
      CACHESIZE      1024
      QUALIFIER      SYSADM
      CURRENTSERVER
      CURRENTDATA    YES
      DEGREE         1
      SQLRULES       DB2
      DISCONNECT     EXPLICIT
      NOREOPT        VARS
      KEEPYNAMIC     NO
DSNT200I . BIND FOR PLAN DSNTEP51 SUCCESSFUL
DSN
  RUN  PROGRAM(DSNTEP2) PLAN(DSNTEP51)          LIB('USER.RUNLIB.LOAD') PARM('/ALIGN(MID)')
DSN
  END
READY
END

```

Figure 115 (Part 2 of 9). Sample Output from Job DSNTJ1P

PAGE 1

***INPUT STATEMENT:

```
GRANT EXECUTE, BIND ON PLAN DSNTPE51 TO PUBLIC;  
DSNT404I SQLCODE = 000, SUCCESSFUL EXECUTION
```

PAGE 1

***INPUT STATEMENT:

```
SELECT EMPNO, FIRSTNME, MIDINIT, LASTNAME,  
       WORKDEPT, PHONENO, HIREDATE, JOB, EDLEVEL,  
       SEX, BIRTHDATE, SALARY, BONUS, COMM,  
       SALARY+BONUS+COMM AS TOTAL_SALARY  
FROM EMP  
ORDER BY TOTAL_SALARY;
```

	EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX
1	000290	JOHN	R	PARKER	E11	4502	1980-05-30	OPERATOR	12	M
2	000310	MAUDE	F	SETRIGHT	E11	3332	1964-09-12	OPERATOR	12	F
3	200310	MICHELLE	F	SPRINGER	E11	3332	1964-09-12	OPERATOR	12	F
4	000260	SYBIL	V	JOHNSON	D21	8953	1975-09-11	CLERK	16	F
5	000300	PHILIP	X	SMITH	E11	2095	1972-06-19	OPERATOR	14	M
6	000210	WILLIAM	T	JONES	D11	0942	1979-04-11	DESIGNER	17	M
7	000250	DANIEL	S	SMITH	D21	0961	1969-10-30	CLERK	15	M
8	000320	RAMLAL	V	MEHTA	E21	9990	1965-07-07	FIELDREP	16	M
9	000190	JAMES	H	WALKER	D11	2986	1974-07-26	DESIGNER	16	M
10	000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07	DESIGNER	17	F
11	000230	JAMES	J	JEFFERSON	D21	4265	1966-11-21	CLERK	14	M
12	000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11	DESIGNER	17	F
13	000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST	16	F
14	000340	JASON	R	GOUNOT	E21	5698	1947-05-05	FIELDREP	16	M
15	200340	ROY	R	ALONZO	E21	5698	1947-05-05	FIELDREP	16	M
16	200170	KIYOSHI		YAMAMOTO	D11	2890	1978-09-15	DESIGNER	16	M
17	000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15	DESIGNER	16	M
18	000150	BRUCE		ADAMSON	D11	4510	1972-02-12	DESIGNER	16	M
19	000330	WING		LEE	E21	2103	1976-02-23	FIELDREP	14	M
20	200330	HELENA		WONG	E21	2103	1976-02-23	FIELDREP	14	F
21	000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER	14	M
22	200280	EILEEN	R	SCHWARTZ	E11	8997	1967-03-24	OPERATOR	17	F
23	000280	ETHEL	R	SCHNEIDER	E11	8997	1967-03-24	OPERATOR	17	F
24	000270	MARIA	L	PEREZ	D21	9001	1980-09-30	CLERK	15	F
25	000200	DAVID		BROWN	D11	4501	1966-03-03	DESIGNER	16	M
26	000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST	18	F
27	200140	KIM	N	NATZ	C01	1793	1976-12-15	ANALYST	18	F
28	200240	ROBERT	M	MONTEVERDE	D21	3780	1979-12-05	CLERK	17	M
29	000240	SALVATORE	M	MARINO	D21	3780	1979-12-05	CLERK	17	M
30	000120	SEAN		O'CONNELL	A00	2167	1963-12-05	CLERK	14	M
31	200120	GREG		ORLANDO	A00	2167	1972-05-05	CLERK	14	M
32	000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER	16	F
33	000220	JENNIFER	K	LUTZ	D11	0672	1968-08-29	DESIGNER	18	F
34	200220	REBA	K	JOHN	D11	0672	1968-08-29	DESIGNER	18	F
35	000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER	16	M
36	000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER	16	F
37	000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER	20	F
38	000050	JOHN	B	GEYER	E01	6789	1949-08-17	MANAGER	16	M
39	000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER	18	M
40	000110	VINCENZO	G	LUCCHESI	A00	3490	1958-05-16	SALESREP	19	M
41	200010	DIAN	J	HEMMINGER	A00	3978	1965-01-01	SALESREP	18	F

Figure 115 (Part 3 of 9). Sample Output from Job DSNTPEJ1P

	BIRTHDATE	SALARY	BONUS	COMM	TOTAL_SALARY
1	1946-07-09	15340.00	300.00	1227.00	16867.00
2	1931-04-21	15900.00	300.00	1272.00	17472.00
3	1931-04-21	15900.00	300.00	1272.00	17472.00
4	1936-10-05	17250.00	300.00	1380.00	18930.00
5	1936-10-27	17750.00	400.00	1420.00	19570.00
6	1953-02-23	18270.00	400.00	1462.00	20132.00
7	1939-11-12	19180.00	400.00	1534.00	21114.00
8	1932-08-11	19950.00	400.00	1596.00	21946.00
9	1952-06-25	20450.00	400.00	1636.00	22486.00
10	1949-02-21	21340.00	500.00	1707.00	23547.00
11	1935-05-30	22180.00	400.00	1774.00	24354.00
12	1955-04-12	22250.00	400.00	1780.00	24430.00
13	1925-09-15	23800.00	500.00	1904.00	26204.00
14	1926-05-17	23840.00	500.00	1907.00	26247.00
15	1926-05-17	23840.00	500.00	1907.00	26247.00
16	1951-01-05	24680.00	500.00	1974.00	27154.00
17	1951-01-05	24680.00	500.00	1974.00	27154.00
18	1947-05-17	25280.00	500.00	2022.00	27802.00
19	1941-07-18	25370.00	500.00	2030.00	27900.00
20	1941-07-18	25370.00	500.00	2030.00	27900.00
21	1956-12-18	26150.00	500.00	2092.00	28742.00
22	1936-03-28	26250.00	500.00	2100.00	28850.00
23	1936-03-28	26250.00	500.00	2100.00	28850.00
24	1953-05-26	27380.00	500.00	2190.00	30070.00
25	1941-05-29	27740.00	600.00	2217.00	30557.00
26	1946-01-19	28420.00	600.00	2274.00	31294.00
27	1946-01-19	28420.00	600.00	2274.00	31294.00
28	1954-03-31	28760.00	600.00	2301.00	31661.00
29	1954-03-31	28760.00	600.00	2301.00	31661.00
30	1942-10-18	29250.00	600.00	2340.00	32190.00
31	1942-10-18	29250.00	600.00	2340.00	32190.00
32	1941-05-15	29750.00	600.00	2380.00	32730.00
33	1948-03-19	29840.00	600.00	2387.00	32827.00
34	1948-03-19	29840.00	600.00	2387.00	32827.00
35	1945-07-07	32250.00	600.00	2580.00	35430.00
36	1953-05-26	36170.00	700.00	2893.00	39763.00
37	1941-05-11	38250.00	800.00	3060.00	42110.00
38	1925-09-15	40175.00	800.00	3214.00	44189.00
39	1948-02-02	41250.00	800.00	3300.00	45350.00
40	1929-11-05	46500.00	900.00	3720.00	51120.00
41	1933-08-14	46500.00	1000.00	4220.00	51720.00

Figure 115 (Part 4 of 9). Sample Output from Job DSNTJ1P

PAGE 3

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX
42_	000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01	PRES	18 F

PAGE 4

BIRTHDATE	SALARY	BONUS	COMM	TOTAL_SALARY
42_	1933-08-14	52750.00	1000.00	4220.00 57970.00

SUCCESSFUL RETRIEVAL OF 42 ROW(S)

PAGE 1

***INPUT STATEMENT:
SELECT * FROM DEPT;

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
1_	A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00
2_	B01	PLANNING	000020	A00
3_	C01	INFORMATION CENTER	000030	A00
4_	D01	DEVELOPMENT CENTER	?	A00
5_	D11	MANUFACTURING SYSTEMS	000060	D01
6_	D21	ADMINISTRATION SYSTEMS	000070	D01
7_	E01	SUPPORT SERVICES	000050	A00
8_	E11	OPERATIONS	000090	E01
9_	E21	SOFTWARE SUPPORT	000100	E01
10_	F22	BRANCH OFFICE F2	?	E01
11_	G22	BRANCH OFFICE G2	?	E01
12_	H22	BRANCH OFFICE H2	?	E01
13_	I22	BRANCH OFFICE I2	?	E01
14_	J22	BRANCH OFFICE J2	?	E01

SUCCESSFUL RETRIEVAL OF 14 ROW(S)

PAGE 1

***INPUT STATEMENT:
SELECT * FROM ACT;

ACTNO	ACTKWD	ACTDESC	
1_	10	MANAGE	MANAGE/ADVISE
2_	20	ECOST	ESTIMATE COST
3_	30	DEFINE	DEFINE SPECS
4_	40	LEADPR	LEAD PROGRAM/DESIGN
5_	50	SPECS	WRITE SPECS
6_	60	LOGIC	DESCRIBE LOGIC
7_	70	CODE	CODE PROGRAMS
8_	80	TEST	TEST PROGRAMS
9_	90	ADMQS	ADM QUERY SYSTEM
10_	100	TEACH	TEACH CLASSES
11_	110	COURSE	DEVELOP COURSES
12_	120	STAFF	PERS AND STAFFING
13_	130	OPERAT	OPER COMPUTER SYS
14_	140	MAINT	MAINT SOFTWARE SYS
15_	150	ADMSYS	ADM OPERATING SYS
16_	160	ADMDB	ADM DATA BASES
17_	170	ADMDC	ADM DATA COMM
18_	180	DOC	DOCUMENT

SUCCESSFUL RETRIEVAL OF 18 ROW(S)

Figure 115 (Part 5 of 9). Sample Output from Job DSNTEJ1P

***INPUT STATEMENT:

SELECT * FROM EMPPROJECT;

	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
1	000130	IF1000	90	1.00	1982-01-01	1982-10-01
2	000130	IF1000	100	0.50	1982-10-01	1983-01-01
3	000140	IF1000	90	0.50	1982-10-01	1983-01-01
4	000030	IF1000	10	0.50	1982-06-01	1983-01-01
5	000030	IF2000	10	0.50	1982-01-01	1983-01-01
6	000140	IF2000	100	1.00	1982-01-01	1982-03-01
7	000140	IF2000	100	0.50	1982-03-01	1982-07-01
8	000140	IF2000	110	0.50	1982-03-01	1982-07-01
9	000140	IF2000	110	0.50	1982-10-01	1983-01-01
10	000010	MA2100	10	0.50	1982-01-01	1982-11-01
11	000110	MA2100	20	1.00	1982-01-01	1982-03-01
12	000020	PL2100	30	1.00	1982-01-01	1982-09-15
13	000010	MA2110	10	1.00	1982-01-01	1982-02-01
14	000220	MA2111	40	1.00	1982-01-01	1983-02-01
15	000200	MA2111	50	1.00	1982-01-01	1982-06-15
16	000200	MA2111	60	1.00	1982-06-15	1983-02-01
17	000150	MA2112	60	1.00	1982-01-01	1982-07-15
18	000150	MA2112	180	1.00	1982-07-15	1983-02-01
19	000170	MA2112	60	1.00	1982-01-01	1983-06-01
20	000170	MA2112	70	1.00	1982-06-01	1983-02-01
21	000190	MA2112	70	1.00	1982-02-01	1982-10-01
22	000190	MA2112	80	1.00	1982-10-01	1983-10-01
23	000160	MA2113	60	1.00	1982-07-15	1983-02-01
24	000170	MA2113	80	1.00	1982-01-01	1983-02-01
25	000180	MA2113	70	1.00	1982-04-01	1982-06-15
26	000210	MA2113	80	0.50	1982-10-01	1983-02-01
27	000210	MA2113	180	0.50	1982-10-01	1983-02-01
28	000010	AD3100	10	0.50	1982-01-01	1982-07-01
29	000070	AD3110	10	1.00	1982-01-01	1983-02-01
30	000230	AD3111	60	1.00	1982-01-01	1982-03-15
31	000230	AD3111	60	0.50	1982-03-15	1982-04-15
32	000230	AD3111	70	0.50	1982-03-15	1982-10-15
33	000230	AD3111	80	0.50	1982-04-15	1982-10-15
34	000230	AD3111	180	1.00	1982-10-15	1983-01-01
35	000240	AD3111	70	1.00	1982-02-15	1982-09-15
36	000240	AD3111	80	1.00	1982-09-15	1983-01-01
37	000250	AD3112	60	1.00	1982-01-01	1982-02-01
38	000250	AD3112	60	0.50	1982-02-01	1982-03-15
39	000250	AD3112	60	0.50	1982-12-01	1983-01-01
40	000250	AD3112	60	1.00	1983-01-01	1983-02-01
41	000250	AD3112	70	0.50	1982-02-01	1982-03-15
42	000250	AD3112	70	1.00	1982-03-15	1982-08-15
43	000250	AD3112	70	0.25	1982-08-15	1982-10-15
44	000250	AD3112	80	0.25	1982-08-15	1982-10-15
45	000250	AD3112	80	0.50	1982-10-15	1982-12-01
46	000250	AD3112	180	0.50	1982-08-15	1983-01-01

Figure 115 (Part 6 of 9). Sample Output from Job DSNTEJ1P

	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
47	000270	AD3113	60	0.50	1982-03-01	1982-04-01
48	000270	AD3113	60	1.00	1982-04-01	1982-09-01
49	000270	AD3113	60	0.25	1982-09-01	1982-10-15
50	000270	AD3113	70	0.75	1982-09-01	1982-10-15
51	000270	AD3113	70	1.00	1982-10-15	1983-02-01
52	000270	AD3113	80	1.00	1982-01-01	1982-03-01
53	000270	AD3113	80	0.50	1982-03-01	1982-04-01
54	000260	AD3113	70	0.50	1982-06-15	1982-07-01
55	000260	AD3113	70	1.00	1982-07-01	1983-02-01
56	000260	AD3113	80	1.00	1982-01-01	1982-03-01
57	000260	AD3113	80	0.50	1982-03-01	1982-04-15
58	000260	AD3113	180	0.50	1982-03-01	1982-04-15
59	000260	AD3113	180	1.00	1982-04-15	1982-06-01
60	000260	AD3113	180	0.50	1982-06-01	1982-07-01
61	000050	OP1000	10	0.25	1982-01-01	1983-02-01
62	000090	OP1010	10	1.00	1982-01-01	1983-02-01
63	000280	OP1010	130	1.00	1982-01-01	1983-02-01
64	000290	OP1010	130	1.00	1982-01-01	1983-02-01
65	000300	OP1010	130	1.00	1982-01-01	1983-02-01
66	000310	OP1010	130	1.00	1982-01-01	1983-02-01
67	000050	OP2010	10	0.75	1982-01-01	1983-02-01
68	000100	OP2010	10	1.00	1982-01-01	1983-02-01
69	000320	OP2011	140	0.75	1982-01-01	1983-02-01
70	000320	OP2011	150	0.25	1982-01-01	1983-02-01
71	000330	OP2012	140	0.25	1982-01-01	1983-02-01
72	000330	OP2012	160	0.75	1982-01-01	1983-02-01
73	000340	OP2013	140	0.50	1982-01-01	1983-02-01
74	000340	OP2013	170	0.50	1982-01-01	1983-02-01

SUCCESSFUL RETRIEVAL OF 74 ROW(S)

PAGE 1

***INPUT STATEMENT:

SELECT * FROM PROJ;

	PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
1	AD3100	ADMIN SERVICES	D01	000010	6.50	1982-01-01	1983-02-01	?
2	AD3110	GENERAL AD SYSTEMS	D21	000070	6.00	1982-01-01	1983-02-01	AD3100
3	AD3111	PAYROLL PROGRAMMING	D21	000230	2.00	1982-01-01	1983-02-01	AD3110
4	AD3112	PERSONNEL PROGRAMMG	D21	000250	1.00	1982-01-01	1983-02-01	AD3110
5	AD3113	ACCOUNT. PROGRAMMING	D21	000270	2.00	1982-01-01	1983-02-01	AD3110
6	IF1000	QUERY SERVICES	C01	000030	2.00	1982-01-01	1983-02-01	?
7	IF2000	USER EDUCATION	C01	000030	1.00	1982-01-01	1983-02-01	?
8	MA2100	WELD LINE AUTOMATION	D01	000010	12.00	1982-01-01	1983-02-01	?
9	MA2110	W L PROGRAMMING	D11	000060	9.00	1982-01-01	1983-02-01	MA2100
10	MA2111	W L PROGRAM DESIGN	D11	000220	2.00	1982-01-01	1982-12-01	MA2110
11	MA2112	W L ROBOT DESIGN	D11	000150	3.00	1982-01-01	1982-12-01	MA2110
12	MA2113	W L PROD CONT PROGS	D11	000160	3.00	1982-02-15	1982-12-01	MA2110
13	OP1000	OPERATION SUPPORT	E01	000050	6.00	1982-01-01	1983-02-01	?
14	OP1010	OPERATION	E11	000090	5.00	1982-01-01	1983-02-01	OP1000
15	OP2000	GEN SYSTEMS SERVICES	E01	000050	5.00	1982-01-01	1983-02-01	?
16	OP2010	SYSTEMS SUPPORT	E21	000100	4.00	1982-01-01	1983-02-01	OP2000
17	OP2011	SCP SYSTEMS SUPPORT	E21	000320	1.00	1982-01-01	1983-02-01	OP2010
18	OP2012	APPLICATIONS SUPPORT	E21	000330	1.00	1982-01-01	1983-02-01	OP2010
19	OP2013	DB/DC SUPPORT	E21	000340	1.00	1982-01-01	1983-02-01	OP2010
20	PL2100	WELD LINE PLANNING	B01	000020	1.00	1982-01-01	1982-09-15	MA2100

SUCCESSFUL RETRIEVAL OF 20 ROW(S)

Figure 115 (Part 7 of 9). Sample Output from Job DSNTJE1P

***INPUT STATEMENT:
SELECT * FROM PROJECT;

	PROJNO	ACTNO	ACSTAFF	ACSTDATE	ACENDATE
1	AD3100	10	0.50	1982-01-01	1982-07-01
2	AD3110	10	1.00	1982-01-01	1983-01-01
3	AD3111	60	0.50	1982-03-15	1982-04-15
4	AD3111	60	0.80	1982-01-01	1982-04-15
5	AD3111	70	0.50	1982-03-15	1982-10-15
6	AD3111	70	1.50	1982-02-15	1982-10-15
7	AD3111	80	1.00	1982-09-15	1983-01-01
8	AD3111	80	1.25	1982-04-15	1983-01-15
9	AD3111	180	1.00	1982-10-15	1983-01-15
10	AD3112	60	0.50	1982-02-01	1982-03-15
11	AD3112	60	0.75	1982-01-01	1982-03-15
12	AD3112	60	0.75	1982-12-01	1983-01-01
13	AD3112	60	1.00	1983-01-01	1983-02-01
14	AD3112	70	0.25	1982-08-15	1982-10-15
15	AD3112	70	0.50	1982-02-01	1982-03-15
16	AD3112	70	0.75	1982-01-01	1982-10-15
17	AD3112	70	1.00	1982-03-15	1982-08-15
18	AD3112	80	0.35	1982-08-15	1982-12-01
19	AD3112	80	0.50	1982-10-15	1982-12-01
20	AD3112	180	0.50	1982-08-15	1983-01-01
21	AD3113	60	0.25	1982-09-01	1982-10-15
22	AD3113	60	0.75	1982-03-01	1982-10-15
23	AD3113	60	1.00	1982-04-01	1982-09-01
24	AD3113	70	0.50	1982-06-15	1982-07-01
25	AD3113	70	0.75	1982-09-01	1982-10-15
26	AD3113	70	1.00	1982-07-01	1983-02-01
27	AD3113	70	1.00	1982-10-15	1983-02-01
28	AD3113	70	1.25	1982-06-01	1982-12-15
29	AD3113	80	0.50	1982-03-01	1982-04-15
30	AD3113	80	1.75	1982-01-01	1982-04-15
31	AD3113	180	0.50	1982-06-01	1982-07-01
32	AD3113	180	0.75	1982-03-01	1982-07-01
33	AD3113	180	1.00	1982-04-15	1982-06-01
34	IF1000	10	0.50	1982-01-01	1983-01-01
35	IF1000	10	0.50	1982-06-01	1983-01-01
36	IF1000	90	0.50	1982-10-01	1983-01-01
37	IF1000	90	1.00	1982-01-01	1983-01-01
38	IF1000	100	0.50	1982-10-01	1983-01-01
39	IF2000	10	0.50	1982-01-01	1983-01-01
40	IF2000	100	0.50	1982-03-01	1982-07-01
41	IF2000	100	0.75	1982-01-01	1982-07-01
42	IF2000	110	0.50	1982-03-01	1982-07-01
43	IF2000	110	0.50	1982-10-01	1983-01-01
44	MA2100	10	0.50	1982-01-01	1982-11-01
45	MA2100	20	1.00	1982-01-01	1982-03-01
46	MA2110	10	1.00	1982-01-01	1983-02-01

Figure 115 (Part 8 of 9). Sample Output from Job DSNTEJ1P

	PROJNO	ACTNO	ACSTAFF	ACSTDATE	ACENDATE
47	MA2111	40	1.00	1982-01-01	1983-02-01
48	MA2111	50	1.00	1982-01-01	1982-06-01
49	MA2111	60	1.00	1982-06-01	1983-02-01
50	MA2111	60	1.00	1982-06-15	1983-02-01
51	MA2112	60	2.00	1982-01-01	1982-07-01
52	MA2112	70	1.00	1982-02-01	1982-10-01
53	MA2112	70	1.00	1982-06-01	1983-02-01
54	MA2112	70	1.50	1982-02-15	1983-02-01
55	MA2112	80	1.00	1982-10-01	1983-10-01
56	MA2112	180	1.00	1982-07-01	1983-02-01
57	MA2112	180	1.00	1982-07-15	1983-02-01
58	MA2113	60	1.00	1982-02-15	1982-09-01
59	MA2113	60	1.00	1982-07-15	1983-02-01
60	MA2113	70	2.00	1982-04-01	1983-12-15
61	MA2113	80	0.50	1982-10-01	1983-02-01
62	MA2113	80	1.00	1982-01-01	1983-02-01
63	MA2113	80	1.50	1982-09-01	1983-02-01
64	MA2113	180	0.50	1982-10-01	1983-01-01
65	OP1000	10	0.25	1982-01-01	1983-02-01
66	OP1010	10	1.00	1982-01-01	1983-02-01
67	OP1010	130	4.00	1982-01-01	1983-02-01
68	OP2000	50	0.75	1982-01-01	1983-02-01
69	OP2010	10	1.00	1982-01-01	1983-02-01
70	OP2011	140	0.75	1982-01-01	1983-02-01
71	OP2011	150	0.25	1982-01-01	1983-02-01
72	OP2012	140	0.25	1982-01-01	1983-02-01
73	OP2012	160	0.75	1982-01-01	1983-02-01
74	OP2013	140	0.50	1982-01-01	1983-02-01
75	OP2013	170	0.50	1982-01-01	1983-02-01
76	PL2100	30	1.00	1982-01-01	1982-09-15
77	PL2100	30	1.00	1982-02-01	1982-09-01

SUCCESSFUL RETRIEVAL OF 77 ROW(S)

Figure 115 (Part 9 of 9). Sample Output from Job DSNTEJ1P

Phase 2

DSNTEJ2A Job

Here is part of the sample output for DSNTEJ2A:

```

READY
  DSN SYSTEM(V51A)
DSN
  BIND PLAN(DSNTIB51) MEM(DSNTIAUL) ACT(REP) ISOLATION(CS)          LIB('USER.DBRMLIB.DATA')
WARNING, ONLY IBM-SUPPLIED PLAN-NAMES SHOULD BEGIN WITH "DSN"
DSNT252I . DSNTBCM1 BIND OPTIONS FOR PLAN DSNTIB51
      ACTION      REPLACE
      OWNER       SYSADM
      VALIDATE    RUN
      ISOLATION   CS
      ACQUIRE    USE
      RELEASE     COMMIT
      EXPLAIN     NO
      DYNAMICRULES RUN
DSNT253I . DSNTBCM1 BIND OPTIONS FOR PLAN DSNTIB51
      NODEFER     PREPARE
      CACHESIZE   1024
      QUALIFIER   SYSADM
      CURRENTSERVER
      CURRENTDATA YES
      DEGREE      1
      SQLRULES    DB2
      DISCONNECT  EXPLICIT
      NOREOPT     VARS
      KEEPYNAMIC  NO
DSNT200I . BIND FOR PLAN DSNTIB51 SUCCESSFUL
DSN
  RUN PROGRAM(DSNTIAD)  PLAN(DSNTIA51)          LIB('USER.RUNLIB.LOAD')
DSN
  END
READY
END
  DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0

```

```

      GRANT EXECUTE ON PLAN DSNTIB51 TO PUBLIC
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
READY
  DELETE 'USRT003.F.DSN8UNLD.SYSREC00'
ENTRY USRT003.F.DSN8UNLD.SYSREC00 NOT FOUND+
** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLEG-42
** ENTRY USRT003.F.DSN8UNLD.SYSREC00 NOT DELETED
LASTCC=8
READY
  DELETE 'USRT003.F.DSN8UNLD.SYSREC01'
ENTRY USRT003.F.DSN8UNLD.SYSREC01 NOT FOUND+
** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLEG-42
** ENTRY USRT003.F.DSN8UNLD.SYSREC01 NOT DELETED
LASTCC=8
READY
  DELETE 'USRT003.F.DSN8UNLD.SYSPUNCH'
ENTRY USRT003.F.DSN8UNLD.SYSPUNCH NOT FOUND+
** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLEG-42
** ENTRY USRT003.F.DSN8UNLD.SYSPUNCH NOT DELETED
LASTCC=8
READY
  DSN SYSTEM(V51A)
DSN
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51) PARM('RC0')          LIB('USER.RUNLIB.LOAD')
DSN
  END

```

Figure 116 (Part 1 of 4). Sample Output from Job DSNTJE2A

```

DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0
      DROP TABLE DSN8510.NEWDEPT
SQL ERROR DURING EXECUTE IMMEDIATE
DSNT408I  SQLCODE = -204, ERROR: DSN8510.NEWDEPT IS AN UNDEFINED NAME
DSNT418I  SQLSTATE = 42704 SQLSTATE RETURN CODE
DSNT415I  SQLERRP = DSNX0FE SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD = -200 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD = X'FFFFFFF38' X'00000000' X'00000000' X'FFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
      INFORMATION

      DROP TABLE DSN8510.NEWPHONE
SQL ERROR DURING EXECUTE IMMEDIATE
DSNT408I  SQLCODE = -204, ERROR: DSN8510.NEWPHONE IS AN UNDEFINED NAME
DSNT418I  SQLSTATE = 42704 SQLSTATE RETURN CODE
DSNT415I  SQLERRP = DSNX0FE SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD = -200 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD = X'FFFFFFF38' X'00000000' X'00000000' X'FFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
      INFORMATION

      DROP DATABASE DSN8D51U
SQL ERROR DURING EXECUTE IMMEDIATE
DSNT408I  SQLCODE = -204, ERROR: DSN8D51U IS AN UNDEFINED NAME
DSNT418I  SQLSTATE = 42704 SQLSTATE RETURN CODE
DSNT415I  SQLERRP = DSNX0FE SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD = -200 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD = X'FFFFFFF38' X'00000000' X'00000000' X'FFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
      INFORMATION

      DROP STOGROUP DSN8G51U
SQL ERROR DURING EXECUTE IMMEDIATE
DSNT408I  SQLCODE = -204, ERROR: DSN8G51U IS AN UNDEFINED NAME
DSNT418I  SQLSTATE = 42704 SQLSTATE RETURN CODE
DSNT415I  SQLERRP = DSNX0FE SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD = -200 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD = X'FFFFFFF38' X'00000000' X'00000000' X'FFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
      INFORMATION

      COMMIT
DSNT400I  SQLCODE = 000, SUCCESSFUL EXECUTION
READY
DSN SYSTEM(V51A)
DSN
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51) LIB('USER.RUNLIB.LOAD')
DSN
END
DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0

      CREATE STOGROUP DSN8G51U
      VOLUMES ("SCR03")
      VCAT DSNCAT
DSNT400I  SQLCODE = 000, SUCCESSFUL EXECUTION

      CREATE DATABASE DSN8D51U
      STOGROUP DSN8G51U
DSNT400I  SQLCODE = 000, SUCCESSFUL EXECUTION

```

Figure 116 (Part 2 of 4). Sample Output from Job DSNTJE2A

```

CREATE TABLE DSN8510.NEWDEPT
      (DEPTNO      CHAR(3)      NOT NULL,
       DEPTNAME    VARCHAR(36)  NOT NULL,
       MGRNO       CHAR(6)      ,
       ADMRDEPT    CHAR(3)      NOT NULL,
       LOCATION    CHAR(16))
      IN DATABASE DSN8D51U
DSNT400I  SQLCODE = 000,  SUCCESSFUL EXECUTION

CREATE TABLE DSN8510.NEWPHONE
      (LASTNAME    VARCHAR(15)  NOT NULL,
       FIRSTNAME   VARCHAR(12)  NOT NULL,
       MIDDLEINITIAL CHAR(1)    NOT NULL,
       PHONENUMBER CHAR(4)      ,
       EMPLOYEEENUMBER CHAR(6)  NOT NULL,
       DEPTNUMBER  CHAR(3)      NOT NULL,
       DEPTNAME    VARCHAR(36)  NOT NULL )
      IN DATABASE DSN8D51U
DSNT400I  SQLCODE = 000,  SUCCESSFUL EXECUTION
READY
DSN SYSTEM(V51A)
DSN
RUN  PROGRAM(DSNTIAUL) PLAN(DSNTIB51) PARS('SQL')      LIB('USER.RUNLIB.LOAD')
DSN
END
DSNT490I  SAMPLE DATA UNLOAD PROGRAM
DSNT505I  DSNTIAUL OPTIONS USED: SQL
DSNT503I  UNLOAD DATA SET SYSPUNCH RECORD LENGTH SET TO 80
DSNT504I  UNLOAD DATA SET SYSPUNCH BLOCK SIZE SET TO 23440

LOCK TABLE DSN8510.DEPT IN SHARE MODE
DSNT400I  SQLCODE = 000,  SUCCESSFUL EXECUTION
DSNT503I  UNLOAD DATA SET SYSREC00 RECORD LENGTH SET TO 68
DSNT504I  UNLOAD DATA SET SYSREC00 BLOCK SIZE SET TO 23460
DSNT495I  SUCCESSFUL UNLOAD          14 ROWS OF TABLE DSN8510.DEPT
DSNT503I  UNLOAD DATA SET SYSREC01 RECORD LENGTH SET TO 85
DSNT504I  UNLOAD DATA SET SYSREC01 BLOCK SIZE SET TO 23460
DSNT495I  SUCCESSFUL UNLOAD          42 ROWS OF TABLE DSN8510.VPHONE
READY
EDIT 'USRT003.F.DSN8UNLD.SYSPUNCH' DATA NONUM
EDIT
CHANGE * 30 /DSN8510.DEPT/DSN8510.NEWDEPT/
CHANGE * 30 /DSN8510.VPHONE/DSN8510.NEWPHONE/
TOP
LIST * 999

```

Figure 116 (Part 3 of 4). Sample Output from Job DSNTTEJ2A

```

LOAD DATA LOG NO INDDN SYSREC00 INTO TABLE
  DSN8510.NEWDEPT
(
  DEPTNO              POSITION(    1    )
  CHAR(                3) ,
  DEPTNAME            POSITION(    4    )
  VARCHAR              ,
  MGRNO               POSITION(   42    )
  CHAR(                6)
  NULLIF(             48)='?',
  ADMRDEPT            POSITION(   49    )
  CHAR(                3) ,
  LOCATION            POSITION(   52    )
  CHAR(               16)
  NULLIF(             68)='?'
)
LOAD DATA LOG NO INDDN SYSREC01 INTO TABLE
  DSN8510.NEWPHONE
(
  LASTNAME             POSITION(    1    )
  VARCHAR              ,
  FIRSTNAME           POSITION(   18    )
  VARCHAR              ,
  MIDDLEINITIAL       POSITION(   32    )
  CHAR(                1) ,
  PHONENUMBER         POSITION(   33    )
  VARCHAR              ,
  EMPLOYEENUMBER      POSITION(   39    )
  CHAR(                6) ,
  DEPTNUMBER          POSITION(   45    )
  CHAR(                3) ,
  DEPTNAME            POSITION(   48    )
  VARCHAR
)
END OF DATA
END SAVE
READY
END
DSNU000I  DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNTEX
DSNU050I  DSNUGUTC - LOAD DATA LOG NO INDDN SYSREC00
DSNU650I  . DSNURWI - INTO TABLE DSN8510.NEWDEPT
DSNU650I  . DSNURWI - (DEPTNO POSITION(1) CHAR(3),
DSNU650I  . DSNURWI - DEPTNAME POSITION(4) VARCHAR,
DSNU650I  . DSNURWI - MGRNO POSITION(42) CHAR(6) NULLIF(48)='?',
DSNU650I  . DSNURWI - ADMRDEPT POSITION(49) CHAR(3),
DSNU650I  . DSNURWI - LOCATION POSITION(52) CHAR(16) NULLIF(68)='?')
DSNU304I  . DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=14 FOR TABLE DSN8510.NEWDEPT
DSNU302I  DSNURILD - (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED=14
DSNU300I  DSNURILD - (RE)LOAD PHASE COMPLETE, ELAPSED TIME=00:00:00
DSNU381I  . DSNUGSRX - TABLESPACE DSN8D51U.NEWDEPT IS IN COPY PENDING
DSNU050I  DSNUGUTC - LOAD DATA LOG NO INDDN SYSREC01
DSNU650I  . DSNURWI - INTO TABLE DSN8510.NEWPHONE
DSNU650I  . DSNURWI - (LASTNAME POSITION(1) VARCHAR,
DSNU650I  . DSNURWI - FIRSTNAME POSITION(18) VARCHAR,
DSNU650I  . DSNURWI - MIDDLEINITIAL POSITION(32) CHAR(1),
DSNU650I  . DSNURWI - PHONENUMBER POSITION(33) VARCHAR,
DSNU650I  . DSNURWI - EMPLOYEENUMBER POSITION(39) CHAR(6),
DSNU650I  . DSNURWI - DEPTNUMBER POSITION(45) CHAR(3),
DSNU650I  . DSNURWI - DEPTNAME POSITION(48) VARCHAR)
DSNU304I  . DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=42 FOR TABLE DSN8510.NEWPHONE
DSNU302I  DSNURILD - (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED=42
DSNU300I  DSNURILD - (RE)LOAD PHASE COMPLETE, ELAPSED TIME=00:00:00
DSNU381I  . DSNUGSRX - TABLESPACE DSN8D51U.NEWPHONE IS IN COPY PENDING
DSNU010I  DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4

```

Figure 116 (Part 4 of 4). Sample Output from Job DSNTEJ2A

DSNTEJ2C Job

Here is the sample output for DSNTEJ2C:

```
1READY
  DSN SYSTEM(V51A)
  DSN
  BIND PLAN(DSN8BH51) MEMBER(DSN8BC3) ACT(REP) ISOLATION(CS)
  WARNING, ONLY IBM-SUPPLIED PLAN-NAMES SHOULD BEGIN WITH "DSN"
  DSNT252I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BH51
          ACTION      REPLACE
          OWNER       SYSADM
          VALIDATE    RUN
          ISOLATION   CS
          ACQUIRE    USE
          RELEASE     COMMIT
          EXPLAIN     NO
          DYNAMICRULES RUN
  DSNT253I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BH51
          NODEFER    PREPARE
          CACHESIZE  1024
          QUALIFIER  SYSADM
          CURRENTSERVER
          CURRENTDATA YES
          DEGREE     1
          SQLRULES   DB2
          DISCONNECT EXPLICIT
          NOREOPT    VARS
          KEEPYNAMIC NO
  DSNT200I . BIND FOR PLAN DSN8BH51 SUCCESSFUL
  DSN
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51)      LIB('USER.RUNLIB.LOAD')
  DSN
  RUN PROGRAM(DSN8BC3) PLAN(DSN8BH51)      LIB('USER.RUNLIB.LOAD')
  DSN
  END
  READY
  END
  DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0
0
          GRANT BIND, EXECUTE ON PLAN DSN8BH51 TO PUBLIC
  DSNT404I SQLCODE = 000, SUCCESSFUL EXECUTION
```

Figure 117 (Part 1 of 3). Sample Output from Job DSNTEJ2C

1----- TELEPHONE DIRECTORY -----						
LAST NAME	FIRST NAME	INITIAL	PHONE NUMBER	EMPLOYEE NUMBER	WORK DEPT	WORK DEPT NAME
OHAAS	CHRISTINE	I	3978	000010	A00	SPIFFY COMPUTER SERVICE DIV.
OLUCCHESI	VINCENZO	G	3490	000110	A00	SPIFFY COMPUTER SERVICE DIV.
OO'CONNELL	SEAN		2167	000120	A00	SPIFFY COMPUTER SERVICE DIV.
OHEMMINGER	DIAN	J	3978	200010	A00	SPIFFY COMPUTER SERVICE DIV.
OORLANDO	GREG		2167	200120	A00	SPIFFY COMPUTER SERVICE DIV.
OTHOMPSON	MICHAEL	L	3476	000020	B01	PLANNING
OKWAN	SALLY	A	4738	000030	C01	INFORMATION CENTER
OQUINTANA	DOLORES	M	4578	000130	C01	INFORMATION CENTER
ONICHOLLS	HEATHER	A	1793	000140	C01	INFORMATION CENTER
ONATZ	KIM	N	1793	200140	C01	INFORMATION CENTER
OSTERN	IRVING	F	6423	000060	D11	MANUFACTURING SYSTEMS
OADAMSON	BRUCE		4510	000150	D11	MANUFACTURING SYSTEMS
OPIANKA	ELIZABETH	R	3782	000160	D11	MANUFACTURING SYSTEMS
OYOSHIMURA	MASATOSHI	J	2890	000170	D11	MANUFACTURING SYSTEMS
OSCOUTTEN	MARILYN	S	1682	000180	D11	MANUFACTURING SYSTEMS
OWALKER	JAMES	H	2986	000190	D11	MANUFACTURING SYSTEMS
OBROWN	DAVID		4501	000200	D11	MANUFACTURING SYSTEMS
OJONES	WILLIAM	T	0942	000210	D11	MANUFACTURING SYSTEMS
OLUTZ	JENNIFER	K	0672	000220	D11	MANUFACTURING SYSTEMS
OYAMAMOTO	KIYOSHI		2890	200170	D11	MANUFACTURING SYSTEMS
OJOHN	REBA	K	0672	200220	D11	MANUFACTURING SYSTEMS
OPULASKI	EVA	D	7831	000070	D21	ADMINISTRATION SYSTEMS
OJEFFERSON	JAMES	J	4265	000230	D21	ADMINISTRATION SYSTEMS
OMARINO	SALVATORE	M	3780	000240	D21	ADMINISTRATION SYSTEMS
OSMITH	DANIEL	S	0961	000250	D21	ADMINISTRATION SYSTEMS
OJOHNSON	SYBIL	V	8953	000260	D21	ADMINISTRATION SYSTEMS
OPEREZ	MARIA	L	9001	000270	D21	ADMINISTRATION SYSTEMS
OMONTEVERDE	ROBERT	M	3780	200240	D21	ADMINISTRATION SYSTEMS
OGEYER	JOHN	B	6789	000050	E01	SUPPORT SERVICES
OHENDERSON	EILEEN	W	5498	000090	E11	OPERATIONS
OSCHNEIDER	ETHEL	R	8997	000280	E11	OPERATIONS
OPARKER	JOHN	R	4502	000290	E11	OPERATIONS
OSMITH	PHILIP	X	2095	000300	E11	OPERATIONS
OSETRIGHT	MAUDE	F	3332	000310	E11	OPERATIONS
OSCHWARTZ	EILEEN	R	8997	200280	E11	OPERATIONS
OSPRINGER	MICHELLE	F	3332	200310	E11	OPERATIONS
OSPENSER	THEODORE	Q	0972	000100	E21	SOFTWARE SUPPORT
OMEHTA	RAMLAL	V	9990	000320	E21	SOFTWARE SUPPORT
OLEE	WING		2103	000330	E21	SOFTWARE SUPPORT
OGOUNOT	JASON	R	5698	000340	E21	SOFTWARE SUPPORT
OWONG	HELENA		2103	200330	E21	SOFTWARE SUPPORT
OALONZO	ROY	R	5698	200340	E21	SOFTWARE SUPPORT

Figure 117 (Part 2 of 3). Sample Output from Job DSNTEJ2C

```

1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
0JONES          WILLIAM      T    0942  000210  D11  MANUFACTURING SYSTEMS
0JOHNSON       SYBIL        V    8953  000260  D21  ADMINISTRATION SYSTEMS
0JOHN          REBA         K    0672  200220  D11  MANUFACTURING SYSTEMS
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
0THOMPSON      MICHAEL      L    3476  000020  B01  PLANNING
0HENDERSON     EILEEN      W    5498  000090  E11  OPERATIONS
0ADAMSON       BRUCE        4510  000150  D11  MANUFACTURING SYSTEMS
0JEFFERSON     JAMES        J    4265  000230  D21  ADMINISTRATION SYSTEMS
0JOHNSON       SYBIL        V    8953  000260  D21  ADMINISTRATION SYSTEMS
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
0SMITH         DANIEL       S    0961  000250  D21  ADMINISTRATION SYSTEMS
0SMITH         PHILIP      X    2095  000300  E11  OPERATIONS
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
0DSN8008I      DSN8BC3-NO EMPLOYEE FOUND IN TABLE
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
0BROWN         DAVID        4501  000200  D11  MANUFACTURING SYSTEMS
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME

0DSN8004I      DSN8BC3-EMPLOYEE SUCCESSFULLY UPDATED

```

Figure 117 (Part 3 of 3). Sample Output from Job DSNTEJ2C

DSNTEJ2D Job

Here is the sample output for DSNTEJ2D:

```
1READY
  DSN SYSTEM(V51A)
  DSN
  BIND PLAN(DSN8BD51) MEMBER(DSN8BD3) ACT(REP) ISOLATION(CS)
  WARNING, ONLY IBM-SUPPLIED PLAN-NAMES SHOULD BEGIN WITH "DSN"
  DSNT252I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BD51
          ACTION      REPLACE
          OWNER       SYSADM
          VALIDATE    RUN
          ISOLATION   CS
          ACQUIRE    USE
          RELEASE     COMMIT
          EXPLAIN     NO
          DYNAMICRULES RUN
  DSNT253I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BD51
          NODEFER     PREPARE
          CACHESIZE   1024
          QUALIFIER   SYSADM
          CURRENTSERVER
          CURRENTDATA YES
          DEGREE      1
          SQLRULES    DB2
          DISCONNECT  EXPLICIT
          NOREOPT     VARS
          KEEPYNAMIC  NO
  DSNT200I . BIND FOR PLAN DSN8BD51 SUCCESSFUL
  DSN
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51)      LIB('USER.RUNLIB.LOAD')
  DSN
  RUN PROGRAM(DSN8BD3) PLAN(DSN8BD51)      LIB('USER.RUNLIB.LOAD')
  DSN
  END
  READY
  END
  DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0
0
          GRANT BIND, EXECUTE ON PLAN DSN8BD51 TO PUBLIC
  DSNT404I SQLCODE = 000, SUCCESSFUL EXECUTION
0 DSN8000I DSN8BD3-REQUEST IS:
          REQUEST LAST NAME      FIRST NAME      EMPNO      NEW XT.NO
  --L      --*      --      --      --      --
```

Figure 118 (Part 1 of 3). Sample Output from DSNTEJ2D

0 ----- TELEPHONE DIRECTORY -----						
LAST NAME	FIRST NAME	INITIAL	PHONE NUMBER	EMPLOYEE NUMBER	WORK DEPT	WORK DEPT NAME
HAAS	CHRISTINE	I	3978	000010	A00	SPIFFY COMPUTER SERVICE DIV.
LUCCHESE	VINCENZO	G	3490	000110	A00	SPIFFY COMPUTER SERVICE DIV.
O'CONNELL	SEAN		2167	000120	A00	SPIFFY COMPUTER SERVICE DIV.
HEMMINGER	DIAN	J	3978	200010	A00	SPIFFY COMPUTER SERVICE DIV.
ORLANDO	GREG		2167	200120	A00	SPIFFY COMPUTER SERVICE DIV.
THOMPSON	MICHAEL	L	3476	000020	B01	PLANNING
KWAN	SALLY	A	4738	000030	C01	INFORMATION CENTER
QUINTANA	DOLORES	M	4578	000130	C01	INFORMATION CENTER
NICHOLLS	HEATHER	A	1793	000140	C01	INFORMATION CENTER
NATZ	KIM	N	1793	200140	C01	INFORMATION CENTER
STERN	IRVING	F	6423	000060	D11	MANUFACTURING SYSTEMS
ADAMSON	BRUCE		4510	000150	D11	MANUFACTURING SYSTEMS
PIANKA	ELIZABETH	R	3782	000160	D11	MANUFACTURING SYSTEMS
YOSHIMURA	MASATOSHI	J	2890	000170	D11	MANUFACTURING SYSTEMS
SCOUTTEN	MARILYN	S	1682	000180	D11	MANUFACTURING SYSTEMS
WALKER	JAMES	H	2986	000190	D11	MANUFACTURING SYSTEMS
BROWN	DAVID		4501	000200	D11	MANUFACTURING SYSTEMS
JONES	WILLIAM	T	0942	000210	D11	MANUFACTURING SYSTEMS
LUTZ	JENNIFER	K	0672	000220	D11	MANUFACTURING SYSTEMS
YAMAMOTO	KIYOSHI		2890	200170	D11	MANUFACTURING SYSTEMS
JOHN	REBA	K	0672	200220	D11	MANUFACTURING SYSTEMS
PULASKI	EVA	D	7831	000070	D21	ADMINISTRATION SYSTEMS
JEFFERSON	JAMES	J	4265	000230	D21	ADMINISTRATION SYSTEMS
MARINO	SALVATORE	M	3780	000240	D21	ADMINISTRATION SYSTEMS
SMITH	DANIEL	S	0961	000250	D21	ADMINISTRATION SYSTEMS
JOHNSON	SYBIL	V	8953	000260	D21	ADMINISTRATION SYSTEMS
PEREZ	MARIA	L	9001	000270	D21	ADMINISTRATION SYSTEMS
MONTEVERDE	ROBERT	M	3780	200240	D21	ADMINISTRATION SYSTEMS
GEYER	JOHN	B	6789	000050	E01	SUPPORT SERVICES
HENDERSON	EILEEN	W	5498	000090	E11	OPERATIONS
SCHNEIDER	ETHEL	R	8997	000280	E11	OPERATIONS
PARKER	JOHN	R	4502	000290	E11	OPERATIONS
SMITH	PHILIP	X	2095	000300	E11	OPERATIONS
SETRIGHT	MAUDE	F	3332	000310	E11	OPERATIONS
SCHWARTZ	EILEEN	R	8997	200280	E11	OPERATIONS
SPRINGER	MICHELLE	F	3332	200310	E11	OPERATIONS
SPENSER	THEODORE	Q	0972	000100	E21	SOFTWARE SUPPORT
MEHTA	RAMLAL	V	9990	000320	E21	SOFTWARE SUPPORT
LEE	WING		2103	000330	E21	SOFTWARE SUPPORT
GOUNOT	JASON	R	5698	000340	E21	SOFTWARE SUPPORT
WONG	HELENA		2103	200330	E21	SOFTWARE SUPPORT
ALONZO	ROY	R	5698	200340	E21	SOFTWARE SUPPORT

Figure 118 (Part 2 of 3). Sample Output from DSNTJE2D

```

- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME FIRST NAME EMPNO NEW XT.NO
  --L --JO% -- -- --
0 ----- TELEPHONE DIRECTORY -----
  LAST NAME FIRST NAME INITIAL PHONE EMPLOYEE WORK WORK
  NUMBER NUMBER DEPT DEPT NAME
  JONES WILLIAM T 0942 000210 D11 MANUFACTURING SYSTEMS
  JOHNSON SYBIL V 8953 000260 D21 ADMINISTRATION SYSTEMS
  JOHN REBA K 0672 200220 D11 MANUFACTURING SYSTEMS
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME FIRST NAME EMPNO NEW XT.NO
  --L --%SON -- -- --
0 ----- TELEPHONE DIRECTORY -----
  LAST NAME FIRST NAME INITIAL PHONE EMPLOYEE WORK WORK
  NUMBER NUMBER DEPT DEPT NAME
  THOMPSON MICHAEL L 3476 000020 B01 PLANNING
  HENDERSON EILEEN W 5498 000090 E11 OPERATIONS
  ADAMSON BRUCE 4510 000150 D11 MANUFACTURING SYSTEMS
  JEFFERSON JAMES J 4265 000230 D21 ADMINISTRATION SYSTEMS
  JOHNSON SYBIL V 8953 000260 D21 ADMINISTRATION SYSTEMS
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME FIRST NAME EMPNO NEW XT.NO
  --L --SMITH -- -- --
0 ----- TELEPHONE DIRECTORY -----
  LAST NAME FIRST NAME INITIAL PHONE EMPLOYEE WORK WORK
  NUMBER NUMBER DEPT DEPT NAME
  SMITH DANIEL S 0961 000250 D21 ADMINISTRATION SYSTEMS
  SMITH PHILIP X 2095 000300 E11 OPERATIONS
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME FIRST NAME EMPNO NEW XT.NO
  --L --BROWN --ALAN -- -- --
0 ----- TELEPHONE DIRECTORY -----
  LAST NAME FIRST NAME INITIAL PHONE EMPLOYEE WORK WORK
  NUMBER NUMBER DEPT DEPT NAME
  DSN8008I DSN8BD3-NO EMPLOYEE FOUND IN TABLE
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME FIRST NAME EMPNO NEW XT.NO
  --L --BROWN --DAVID -- -- --
0 ----- TELEPHONE DIRECTORY -----
  LAST NAME FIRST NAME INITIAL PHONE EMPLOYEE WORK WORK
  NUMBER NUMBER DEPT DEPT NAME
  BROWN DAVID 4501 000200 D11 MANUFACTURING SYSTEMS
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME FIRST NAME EMPNO NEW XT.NO
  --U -- -- --000230--4265 --
DSN8004I DSN8BD3-EMPLOYEE SUCCESSFULLY UPDATED

```

Figure 118 (Part 3 of 3). Sample Output from DSNTJEJ2D

DSNTEJ2E Job

Here is the sample output from DSNTEJ2E:

```
1READY
  DSN SYSTEM(V51A)
  DSN
  BIND PLAN(DSN8BE51) MEMBER(DSN8BE3) ACT(REP) ISOLATION(CS)
  WARNING, ONLY IBM-SUPPLIED PLAN-NAMES SHOULD BEGIN WITH "DSN"
  DSNT252I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BE51
          ACTION      ADD
          OWNER        SYSADM
          VALIDATE     RUN
          ISOLATION    CS
          ACQUIRE     USE
          RELEASE      COMMIT
          EXPLAIN      NO
          DYNAMICRULES RUN
  DSNT253I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BE51
          NODEFER     PREPARE
          CACHESIZE   1024
          QUALIFIER   SYSADM
          CURRENTSERVER
          CURRENTDATA YES
          DEGREE      1
          SQLRULES    DB2
          DISCONNECT  EXPLICIT
          NOREOPT     VARS
          KEEPYNAMIC  NO
  DSNT200I . BIND FOR PLAN DSN8BE51 SUCCESSFUL
  DSN
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51)      LIB('USER.RUNLIB.LOAD')
  DSN
  RUN PROGRAM(DSN8BE3) PLAN(DSN8BE51)      LIB('USER.RUNLIB.LOAD')
  DSN
  END
  READY
  END
  DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0
0
          GRANT BIND, EXECUTE ON PLAN DSN8BE51 TO PUBLIC
  DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
0 DSN8000I DSN8BD3-REQUEST IS:
          REQUEST LAST NAME      FIRST NAME      EMPNO      NEW XT.NO
  --L      --*      --      --      --      --
```

Figure 119 (Part 1 of 3). Sample Output from DSNTEJ2E

0 -----
 TELEPHONE DIRECTORY

LAST NAME	FIRST NAME	INITIAL	PHONE NUMBER	EMPLOYEE NUMBER	WORK DEPT	WORK DEPT NAME
HAAS	CHRISTINE	I	3978	000010	A00	SPIFFY COMPUTER SERVICE DIV.
LUCCHESI	VINCENZO	G	3490	000110	A00	SPIFFY COMPUTER SERVICE DIV.
O'CONNELL	SEAN		2167	000120	A00	SPIFFY COMPUTER SERVICE DIV.
HEMMINGER	DIAN	J	3978	200010	A00	SPIFFY COMPUTER SERVICE DIV.
ORLANDO	GREG		2167	200120	A00	SPIFFY COMPUTER SERVICE DIV.
THOMPSON	MICHAEL	L	3476	000020	B01	PLANNING
KWAN	SALLY	A	4738	000030	C01	INFORMATION CENTER
QUINTANA	DOLORES	M	4578	000130	C01	INFORMATION CENTER
NICHOLLS	HEATHER	A	1793	000140	C01	INFORMATION CENTER
NATZ	KIM	N	1793	200140	C01	INFORMATION CENTER
STERN	IRVING	F	6423	000060	D11	MANUFACTURING SYSTEMS
ADAMSON	BRUCE		4510	000150	D11	MANUFACTURING SYSTEMS
PIANKA	ELIZABETH	R	3782	000160	D11	MANUFACTURING SYSTEMS
YOSHIMURA	MASATOSHI	J	2890	000170	D11	MANUFACTURING SYSTEMS
SCOUTTEN	MARILYN	S	1682	000180	D11	MANUFACTURING SYSTEMS
WALKER	JAMES	H	2986	000190	D11	MANUFACTURING SYSTEMS
BROWN	DAVID		4501	000200	D11	MANUFACTURING SYSTEMS
JONES	WILLIAM	T	0942	000210	D11	MANUFACTURING SYSTEMS
LUTZ	JENNIFER	K	0672	000220	D11	MANUFACTURING SYSTEMS
YAMAMOTO	KIYOSHI		2890	200170	D11	MANUFACTURING SYSTEMS
JOHN	REBA	K	0672	200220	D11	MANUFACTURING SYSTEMS
PULASKI	EVA	D	7831	000070	D21	ADMINISTRATION SYSTEMS
JEFFERSON	JAMES	J	4265	000230	D21	ADMINISTRATION SYSTEMS
MARINO	SALVATORE	M	3780	000240	D21	ADMINISTRATION SYSTEMS
SMITH	DANIEL	S	0961	000250	D21	ADMINISTRATION SYSTEMS
JOHNSON	SYBIL	V	8953	000260	D21	ADMINISTRATION SYSTEMS
PEREZ	MARIA	L	9001	000270	D21	ADMINISTRATION SYSTEMS
MONTEVERDE	ROBERT	M	3780	200240	D21	ADMINISTRATION SYSTEMS
GEYER	JOHN	B	6789	000050	E01	SUPPORT SERVICES
HENDERSON	EILEEN	W	5498	000090	E11	OPERATIONS
SCHNEIDER	ETHEL	R	8997	000280	E11	OPERATIONS
PARKER	JOHN	R	4502	000290	E11	OPERATIONS
SMITH	PHILIP	X	2095	000300	E11	OPERATIONS
SETRIGHT	MAUDE	F	3332	000310	E11	OPERATIONS
SCHWARTZ	EILEEN	R	8997	200280	E11	OPERATIONS
SPRINGER	MICHELLE	F	3332	200310	E11	OPERATIONS
SPENSER	THEODORE	Q	0972	000100	E21	SOFTWARE SUPPORT
MEHTA	RAMLAL	V	9990	000320	E21	SOFTWARE SUPPORT
LEE	WING		2103	000330	E21	SOFTWARE SUPPORT
GOUNOT	JASON	R	5698	000340	E21	SOFTWARE SUPPORT
WONG	HELENA		2103	200330	E21	SOFTWARE SUPPORT
ALONZO	ROY	R	5698	200340	E21	SOFTWARE SUPPORT

- DSN8000I DSN8BD3-REQUEST IS:
 REQUEST LAST NAME FIRST NAME EMPNO NEW XT.NO
 --L --J0% -- -- -- --

Figure 119 (Part 2 of 3). Sample Output from DSNTJE

```

0 -----
TELEPHONE DIRECTORY
-----
LAST NAME      FIRST NAME    INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER      NUMBER        DEPT DEPT NAME
JONES          WILLIAM      T   0942   000210   D11  MANUFACTURING SYSTEMS
JOHNSON        SYBIL        V   8953   000260   D21  ADMINISTRATION SYSTEMS
JOHN           REBA         K   0672   200220   D11  MANUFACTURING SYSTEMS
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME    FIRST NAME    EMPNO    NEW XT.NO
  --L     --%SON      --           --      --      --
0 -----
TELEPHONE DIRECTORY
-----
LAST NAME      FIRST NAME    INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER      NUMBER        DEPT DEPT NAME
THOMPSON       MICHAEL      L   3476   000020   B01  PLANNING
HENDERSON      EILEEN      W   5498   000090   E11  OPERATIONS
ADAMSON        BRUCE        4510  000150   D11  MANUFACTURING SYSTEMS
JEFFERSON      JAMES        J   4265   000230   D21  ADMINISTRATION SYSTEMS
JOHNSON        SYBIL        V   8953   000260   D21  ADMINISTRATION SYSTEMS
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME    FIRST NAME    EMPNO    NEW XT.NO
  --L     --SMITH     --           --      --      --
0 -----
TELEPHONE DIRECTORY
-----
LAST NAME      FIRST NAME    INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER      NUMBER        DEPT DEPT NAME
SMITH          DANIEL      S   0961   000250   D21  ADMINISTRATION SYSTEMS
SMITH          PHILIP      X   2095   000300   E11  OPERATIONS
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME    FIRST NAME    EMPNO    NEW XT.NO
  --L     --BROWN     --ALAN      --      --      --
0 -----
TELEPHONE DIRECTORY
-----
LAST NAME      FIRST NAME    INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER      NUMBER        DEPT DEPT NAME
DSN8008I DSN8BD3-NO EMPLOYEE FOUND IN TABLE
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME    FIRST NAME    EMPNO    NEW XT.NO
  --L     --BROWN     --DAVID     --      --      --
0 -----
TELEPHONE DIRECTORY
-----
LAST NAME      FIRST NAME    INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER      NUMBER        DEPT DEPT NAME
BROWN         DAVID        4501  000200   D11  MANUFACTURING SYSTEMS
- DSN8000I DSN8BD3-REQUEST IS:
  REQUEST LAST NAME    FIRST NAME    EMPNO    NEW XT.NO
  --U     --           --      --000230 --4265  --
DSN8004I DSN8BD3-EMPLOYEE SUCCESSFULLY UPDATED

```

Figure 119 (Part 3 of 3). Sample Output from DSNTJE2E

DSNTEJ2F Job

Here is the sample output from DSNTEJ2F:

```
1READY
  DSN SYSTEM(V51A)
  DSN
  BIND PLAN(DSN8BF51) MEMBER(DSN8BF3) ACT(REP) ISO(CS)
  WARNING, ONLY IBM-SUPPLIED PLAN-NAMES SHOULD BEGIN WITH "DSN"
  DSNT252I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BF51
            ACTION      REPLACE
            OWNER       SYSADM
            VALIDATE    RUN
            ISOLATION   CS
            ACQUIRE    USE
            RELEASE     COMMIT
            EXPLAIN     NO
            DYNAMICRULES RUN
  DSNT253I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BF51
            NODEFER     PREPARE
            CACHESIZE   1024
            QUALIFIER   SYSADM
            CURRENTSERVER
            CURRENTDATA YES
            DEGREE      1
            SQLRULES    DB2
            DISCONNECT  EXPLICIT
            NOREOPT     VARS
            KEEPYNAMIC  NO
  DSNT200I . BIND FOR PLAN DSN8BF51 SUCCESSFUL
  DSN
  RUN  PROGRAM(DSN8BF3) PLAN(DSN8BF51)      LIB('USER.RUNLIB.LOAD')
  DSN
  RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA51)      LIB('USER.RUNLIB.LOAD')
  DSN
  END
  READY
  END
1DSN8000I: DSN8BF3-REQUEST IS:
```

```
      REQUEST  LAST NAME      FIRST NAME  EMPNO  NEW XT.NO
--L      --*      --      --      --      --
```

Figure 120 (Part 1 of 3). Sample Output from DSNTEJ2F

```

0----- TELEPHONE DIRECTORY -----
OLAST NAME      FIRST NAME  INITIAL  PHONE  EMPLOYEE  WORK WORK
                NUMBER    NUMBER   DEPT  DEPT  NAME
HAAS            CHRISTINE  I    3978   000010   A00  SPIFFY COMPUTER SERVICE DIV.
LUCCHESI       VINCENZO  G    3490   000110   A00  SPIFFY COMPUTER SERVICE DIV.
O'CONNELL      SEAN      2167   000120   A00  SPIFFY COMPUTER SERVICE DIV.
HEMMINGER      DIAN      J    3978   200010   A00  SPIFFY COMPUTER SERVICE DIV.
ORLANDO        GREG      2167   200120   A00  SPIFFY COMPUTER SERVICE DIV.
THOMPSON       MICHAEL   L    3476   000020   B01  PLANNING
KWAN           SALLY     A    4738   000030   C01  INFORMATION CENTER
QUINTANA       DOLORES   M    4578   000130   C01  INFORMATION CENTER
NICHOLLS       HEATHER   A    1793   000140   C01  INFORMATION CENTER
NATZ           KIM      N    1793   200140   C01  INFORMATION CENTER
STERN          IRVING    F    6423   000060   D11  MANUFACTURING SYSTEMS
ADAMSON        BRUCE     4510   000150   D11  MANUFACTURING SYSTEMS
PIANKA        ELIZABETH R    3782   000160   D11  MANUFACTURING SYSTEMS
YOSHIMURA     MASATOSHI J    2890   000170   D11  MANUFACTURING SYSTEMS
SCOUTTEN       MARILYN   S    1682   000180   D11  MANUFACTURING SYSTEMS
WALKER         JAMES     H    2986   000190   D11  MANUFACTURING SYSTEMS
BROWN          DAVID     4501   000200   D11  MANUFACTURING SYSTEMS
JONES          WILLIAM   T    0942   000210   D11  MANUFACTURING SYSTEMS
LUTZ           JENNIFER K    0672   000220   D11  MANUFACTURING SYSTEMS
YAMAMOTO       KIYOSHI   2890   200170   D11  MANUFACTURING SYSTEMS
JOHN           REBA      K    0672   200220   D11  MANUFACTURING SYSTEMS
PULASKI        EVA       D    7831   000070   D21  ADMINISTRATION SYSTEMS
JEFFERSON      JAMES     J    4265   000230   D21  ADMINISTRATION SYSTEMS
MARINO         SALVATORE M    3780   000240   D21  ADMINISTRATION SYSTEMS
SMITH          DANIEL    S    0961   000250   D21  ADMINISTRATION SYSTEMS
JOHNSON        SYBIL     V    8953   000260   D21  ADMINISTRATION SYSTEMS
PEREZ          MARIA     L    9001   000270   D21  ADMINISTRATION SYSTEMS
MONTEVERDE     ROBERT    M    3780   200240   D21  ADMINISTRATION SYSTEMS
GEYER          JOHN      B    6789   000050   E01  SUPPORT SERVICES
HENDERSON      EILEEN    W    5498   000090   E11  OPERATIONS
SCHNEIDER      ETHEL     R    8997   000280   E11  OPERATIONS
PARKER         JOHN      R    4502   000290   E11  OPERATIONS
SMITH          PHILIP    X    2095   000300   E11  OPERATIONS
SETRIGHT       MAUDE     F    3332   000310   E11  OPERATIONS
SCHWARTZ       EILEEN    R    8997   200280   E11  OPERATIONS
SPRINGER       MICHELLE  F    3332   200310   E11  OPERATIONS
SPENSER        THEODORE  Q    0972   000100   E21  SOFTWARE SUPPORT
MEHTA          RAMLAL    V    9990   000320   E21  SOFTWARE SUPPORT
LEE            WING      2103   000330   E21  SOFTWARE SUPPORT
GOUNOT         JASON     R    5698   000340   E21  SOFTWARE SUPPORT
WONG           HELENA    2103   200330   E21  SOFTWARE SUPPORT
ALONZO         ROY       R    5698   200340   E21  SOFTWARE SUPPORT
1DSN80001: DSN8BF3-REQUEST IS:

    REQUEST  LAST NAME    FIRST NAME  EMPNO  NEW XT.NO
--L      --J0%      --          --      --

```

Figure 120 (Part 2 of 3). Sample Output from DSNTEJ2F


```

1
0----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME  INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER  NUMBER  DEPT DEPT NAME

JONES           WILLIAM    T   0942   000210  D11  MANUFACTURING SYSTEMS
JOHNSON         SYBIL      V   8953   000260  D21  ADMINISTRATION SYSTEMS
JOHN            REBA       K   0672   200220  D11  MANUFACTURING SYSTEMS

```

1DSN8000I: DSN8BF3-REQUEST IS:

```

      REQUEST LAST NAME      FIRST NAME  EMPNO  NEW XT.NO
--L      --%SON          --          --      --      --

```

```

1
0----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME  INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER  NUMBER  DEPT DEPT NAME

THOMPSON        MICHAEL    L   3476   000020  B01  PLANNING
HENDERSON       EILEEN     W   5498   000090  E11  OPERATIONS
ADAMSON         BRUCE      4510   000150  D11  MANUFACTURING SYSTEMS
JEFFERSON       JAMES      J   4265   000230  D21  ADMINISTRATION SYSTEMS
JOHNSON         SYBIL      V   8953   000260  D21  ADMINISTRATION SYSTEMS

```

1DSN8000I: DSN8BF3-REQUEST IS:

```

      REQUEST LAST NAME      FIRST NAME  EMPNO  NEW XT.NO
--L      --SMITH          --          --      --      --

```

```

1
0----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME  INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER  NUMBER  DEPT DEPT NAME

SMITH           DANIEL     S   0961   000250  D21  ADMINISTRATION SYSTEMS
SMITH           PHILIP     X   2095   000300  E11  OPERATIONS

```

1DSN8000I: DSN8BF3-REQUEST IS:

```

      REQUEST LAST NAME      FIRST NAME  EMPNO  NEW XT.NO
--L      --BROWN          --ALAN      --      --      --

```

```

1
0----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME  INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER  NUMBER  DEPT DEPT NAME

```

0DSN8008I: DSN8BF3-NO EMPLOYEE FOUND IN TABLE

1DSN8000I: DSN8BF3-REQUEST IS:

```

      REQUEST LAST NAME      FIRST NAME  EMPNO  NEW XT.NO
--L      --BROWN          --DAVID     --      --      --

```

```

1
0----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME  INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER  NUMBER  DEPT DEPT NAME

BROWN           DAVID      4501   000200  D11  MANUFACTURING SYSTEMS

```

1DSN8000I: DSN8BF3-REQUEST IS:

```

      REQUEST LAST NAME      FIRST NAME  EMPNO  NEW XT.NO
--U      --          --          --000230--4265  --

```

```

1
0----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME  INITIAL PHONE  EMPLOYEE WORK WORK
                NUMBER  NUMBER  DEPT DEPT NAME

```

0DSN8004I: DSN8BF3-EMPLOYEE SUCCESSFULLY UPDATED

1

0DSN8051I: DSN8BF3-PROGRAM ENDED
DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0

0

GRANT BIND, EXECUTE ON PLAN DSN8BF51 TO PUBLIC
DSNT404I SQLCODE = 000, SUCCESSFUL EXECUTION

Figure 120 (Part 3 of 3). Sample Output from DSNTJ2F

DSNTEJ2P Job

Here is the sample output from DSNTEJ2P:

```
1READY
  DSN SYSTEM(V51A)
  DSN
  BIND PLAN(DSN8BP51) MEMBER(DSN8BP3) ACT(REP) ISO(CS)
  WARNING, ONLY IBM-SUPPLIED PLAN-NAMES SHOULD BEGIN WITH "DSN"
  DSNT252I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BP51
          ACTION      REPLACE
          OWNER        SYSADM
          VALIDATE     RUN
          ISOLATION    CS
          ACQUIRE     USE
          RELEASE      COMMIT
          EXPLAIN      NO
          DYNAMICRULES RUN
  DSNT253I . DSNTBCM1 BIND OPTIONS FOR PLAN DSN8BP51
          NODEFER     PREPARE
          CACHESIZE   1024
          QUALIFIER   SYSADM
          CURRENTSERVER
          CURRENTDATA YES
          DEGREE      1
          SQLRULES    DB2
          DISCONNECT  EXPLICIT
          NOREOPT     VARS
          KEEPYNAMIC  NO
  DSNT200I . BIND FOR PLAN DSN8BP51 SUCCESSFUL
  DSN
  RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA51)      LIB('USER.RUNLIB.LOAD')
  DSN
  END
  READY
  END
  DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0
0
          GRANT BIND, EXECUTE ON PLAN DSN8BP51 TO PUBLIC
  DSNT404I SQLCODE = 000, SUCCESSFUL EXECUTION
1READY
  DSN SYSTEM(V51A)
  DSN
  RUN  PROGRAM(DSN8BP3) PLAN(DSN8BP51)      LIB('USER.RUNLIB.LOAD')
  DSN
  END
  READY
  END
1
```

Figure 121 (Part 1 of 3). Sample Output from Job DSNTEJ2P

1----- TELEPHONE DIRECTORY -----						
LAST NAME	FIRST NAME	INITIAL	PHONE NUMBER	EMPLOYEE NUMBER	WORK DEPT	WORK DEPT NAME
HAAS	CHRISTINE	I	3978	000010	A00	SPIFFY COMPUTER SERVICE DIV.
LUCCHESI	VINCENZO	G	3490	000110	A00	SPIFFY COMPUTER SERVICE DIV.
O'CONNELL	SEAN		2167	000120	A00	SPIFFY COMPUTER SERVICE DIV.
HEMMINGER	DIAN	J	3978	200010	A00	SPIFFY COMPUTER SERVICE DIV.
ORLANDO	GREG		2167	200120	A00	SPIFFY COMPUTER SERVICE DIV.
THOMPSON	MICHAEL	L	3476	000020	B01	PLANNING
KWAN	SALLY	A	4738	000030	C01	INFORMATION CENTER
QUINTANA	DOLORES	M	4578	000130	C01	INFORMATION CENTER
NICHOLLS	HEATHER	A	1793	000140	C01	INFORMATION CENTER
NATZ	KIM	N	1793	200140	C01	INFORMATION CENTER
STERN	IRVING	F	6423	000060	D11	MANUFACTURING SYSTEMS
ADAMSON	BRUCE		4510	000150	D11	MANUFACTURING SYSTEMS
PIANKA	ELIZABETH	R	3782	000160	D11	MANUFACTURING SYSTEMS
YOSHIMURA	MASATOSHI	J	2890	000170	D11	MANUFACTURING SYSTEMS
SCOUTTEN	MARILYN	S	1682	000180	D11	MANUFACTURING SYSTEMS
WALKER	JAMES	H	2986	000190	D11	MANUFACTURING SYSTEMS
BROWN	DAVID		4501	000200	D11	MANUFACTURING SYSTEMS
JONES	WILLIAM	T	0942	000210	D11	MANUFACTURING SYSTEMS
LUTZ	JENNIFER	K	0672	000220	D11	MANUFACTURING SYSTEMS
YAMAMOTO	KIYOSHI		2890	200170	D11	MANUFACTURING SYSTEMS
JOHN	REBA	K	0672	200220	D11	MANUFACTURING SYSTEMS
PULASKI	EVA	D	7831	000070	D21	ADMINISTRATION SYSTEMS
JEFFERSON	JAMES	J	4265	000230	D21	ADMINISTRATION SYSTEMS
MARINO	SALVATORE	M	3780	000240	D21	ADMINISTRATION SYSTEMS
SMITH	DANIEL	S	0961	000250	D21	ADMINISTRATION SYSTEMS
JOHNSON	SYBIL	V	8953	000260	D21	ADMINISTRATION SYSTEMS
PEREZ	MARIA	L	9001	000270	D21	ADMINISTRATION SYSTEMS
MONTEVERDE	ROBERT	M	3780	200240	D21	ADMINISTRATION SYSTEMS
GEYER	JOHN	B	6789	000050	E01	SUPPORT SERVICES
HENDERSON	EILEEN	W	5498	000090	E11	OPERATIONS
SCHNEIDER	ETHEL	R	8997	000280	E11	OPERATIONS
PARKER	JOHN	R	4502	000290	E11	OPERATIONS
SMITH	PHILIP	X	2095	000300	E11	OPERATIONS
SETRIGHT	MAUDE	F	3332	000310	E11	OPERATIONS
SCHWARTZ	EILEEN	R	8997	200280	E11	OPERATIONS
SPRINGER	MICHELLE	F	3332	200310	E11	OPERATIONS
SPENSER	THEODORE	Q	0972	000100	E21	SOFTWARE SUPPORT
MEHTA	RAMLAL	V	9990	000320	E21	SOFTWARE SUPPORT
LEE	WING		2103	000330	E21	SOFTWARE SUPPORT
GOUNOT	JASON	R	5698	000340	E21	SOFTWARE SUPPORT
WONG	HELENA		2103	200330	E21	SOFTWARE SUPPORT
ALONZO	ROY	R	5698	200340	E21	SOFTWARE SUPPORT

1----- TELEPHONE DIRECTORY -----						
LAST NAME	FIRST NAME	INITIAL	PHONE NUMBER	EMPLOYEE NUMBER	WORK DEPT	WORK DEPT NAME
JONES	WILLIAM	T	0942	000210	D11	MANUFACTURING SYSTEMS
JOHNSON	SYBIL	V	8953	000260	D21	ADMINISTRATION SYSTEMS
JOHN	REBA	K	0672	200220	D11	MANUFACTURING SYSTEMS

Figure 121 (Part 2 of 3). Sample Output from Job DSNTJ2P

```

1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
  THOMPSON      MICHAEL      L    3476   000020  B01  PLANNING
  HENDERSON     EILEEN      W    5498   000090  E11  OPERATIONS
  ADAMSON       BRUCE       4510   000150  D11  MANUFACTURING SYSTEMS
  JEFFERSON     JAMES       J    4265   000230  D21  ADMINISTRATION SYSTEMS
  JOHNSON       SYBIL       V    8953   000260  D21  ADMINISTRATION SYSTEMS
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
  SMITH         DANIEL      S    0961   000250  D21  ADMINISTRATION SYSTEMS
  SMITH         PHILIP     X    2095   000300  E11  OPERATIONS
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
0DSN8000I      DSN8BP3-NO EMPLOYEE FOUND IN TABLE
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME
  BROWN         DAVID      4501   000200  D11  MANUFACTURING SYSTEMS
1----- TELEPHONE DIRECTORY -----
0LAST NAME      FIRST NAME    INITIAL  PHONE  EMPLOYEE WORK WORK
                NUMBER     NUMBER  DEPT  DEPT NAME

0DSN8004I      DSN8BP3-EMPLOYEE SUCCESSFULLY UPDATED

```

Figure 121 (Part 3 of 3). Sample Output from Job DSNTEJ2P

Phase 3 (DSNTEJ3P Job)

Here is the sample output for DSNTEJ3P:

```
1READY
  DSN SYSTEM(V51A)
DSN
  BIND PLAN(DSN8SP51) MEMBER(DSN8SP3) ACT(REP) ISO(CS)
WARNING, ONLY IBM-SUPPLIED PLAN-NAMES SHOULD BEGIN WITH "DSN"
DSNT252I . DSNBDCM1 BIND OPTIONS FOR PLAN DSN8SP51
          ACTION      REPLACE
          OWNER       SYSADM
          VALIDATE    RUN
          ISOLATION   CS
          ACQUIRE    USE
          RELEASE     COMMIT
          EXPLAIN     NO
          DYNAMICRULES RUN
DSNT253I . DSNBDCM1 BIND OPTIONS FOR PLAN DSN8SP51
          NODEFER     PREPARE
          CACHESIZE   1024
          QUALIFIER   SYSADM
          CURRENTSERVER
          CURRENTDATA YES
          DEGREE      1
          SQLRULES   DB2
          DISCONNECT EXPLICIT
          NOREOPT    VARS
          KEEPYNAMIC NO
DSNT200I . BIND FOR PLAN DSN8SP51 SUCCESSFUL
DSN
  RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA51)      LIB('USER.RUNLIB.LOAD')
DSN
  END
READY
END
  DSNTIAD - SAMPLE DYNAMIC SQL PROGRAM 2.0
0
          GRANT BIND, EXECUTE ON PLAN DSN8SP51 TO PUBLIC
DSNT404I SQLCODE = 000, SUCCESSFUL EXECUTION
1READY
  ALLOC DD(SYSIN) SHR          DSN('USRT003.F.NEW.SDSNSAMP(DSNTESA)')
READY
  DSN SYSTEM(V51A)
DSN
  RUN  PROGRAM(DSNTEP2) PLAN(DSNTEP51)      LIB('USER.RUNLIB.LOAD')
DSN
  END
READY
  ALLOC DD(SYSIN) SHR REUSE     DSN('USRT003.F.NEW.SDSNSAMP(DSNTESC) 'USRT003.F.SDSNSAMP(DSNTESA)')
READY
  DSN SYSTEM(V51A)
DSN
  RUN  PROGRAM(DSNTEP2) PLAN(DSNTEP51)      LIB('USER.RUNLIB.LOAD')
DSN
  END
READY
END
1
```

Figure 122 (Part 1 of 23). Sample Output from Job DSNTEJ3P

```

1PAGE 1
***INPUT STATEMENT:
CREATE TABLESPACE TEMPSPAC;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
CREATE SUCCESSFUL
1PAGE 1
***INPUT STATEMENT:
CREATE TABLE TEMPTABLE
(NAME CHAR(8),
MIDTERM SMALLINT,
FINAL SMALLINT,
SUM SMALLINT,
GRADE CHAR(1) )
IN TEMPSPAC;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
CREATE SUCCESSFUL
1PAGE 1
***INPUT STATEMENT:
INSERT INTO TEMPTABLE ( NAME, MIDTERM, FINAL )
VALUES ( 'JOE', 55, 68 );
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000001' X'FFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL INSERT OF 1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
INSERT INTO TEMPTABLE ( NAME, MIDTERM, FINAL )
VALUES ( 'SUE', 83, 87 );
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000001' X'FFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL INSERT OF 1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
INSERT INTO TEMPTABLE ( NAME, MIDTERM, FINAL )
VALUES ( 'KEN', 50, 65 );
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000001' X'FFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL INSERT OF 1 ROW(S)

```

Figure 122 (Part 2 of 23). Sample Output from Job DSNTJ3P

```

1PAGE 1
***INPUT STATEMENT:
  INSERT INTO TEMPTABLE ( NAME, MIDTERM, FINAL )
    VALUES ( 'JAN', 92, 95 );
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD  = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD  = X'00000000' X'00000000' X'00000001' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL INSERT OF          1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
  INSERT INTO TEMPTABLE ( NAME, MIDTERM, FINAL )
    VALUES ( 'LES', 67, 75 );
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD  = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD  = X'00000000' X'00000000' X'00000001' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL INSERT OF          1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
  INSERT INTO TEMPTABLE ( NAME, MIDTERM, FINAL )
    VALUES ( 'SALLY', 75, 78 );
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD  = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD  = X'00000000' X'00000000' X'00000001' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL INSERT OF          1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
  SELECT AVG(FINAL) FROM TEMPTABLE;

```

+-----+	
+-----+	
1_	78
+-----+	

```

0SUCCESSFUL RETRIEVAL OF          1 ROW(S)

```

Figure 122 (Part 3 of 23). Sample Output from Job DSNTJ3P

```

1PAGE 1
***INPUT STATEMENT:
SELECT AVG(MIDTERM) FROM TEMPTABLE;

```

1_	70

0SUCCESSFUL RETRIEVAL OF 1 ROW(S)

```

1PAGE 1
***INPUT STATEMENT:
UPDATE TEMPTABLE SET SUM = .6 * FINAL + .4 * MIDTERM;
SQLWARNING ON UPDATE COMMAND, PREPARE FUNCTION
RESULT OF SQL STATEMENT:

```

```

DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD = 0 0 6 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000006' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION

```

SUCCESSFUL UPDATE OF 6 ROW(S)

```

1PAGE 1
***INPUT STATEMENT:
SELECT AVG(SUM) , MIN(SUM), MAX(SUM) FROM TEMPTABLE;

```

1_	74	59	93

0SUCCESSFUL RETRIEVAL OF 1 ROW(S)

Figure 122 (Part 4 of 23). Sample Output from Job DSNTEJ3P


```

1PAGE 1
***INPUT STATEMENT:
  UPDATE TEMPTABLE SET GRADE = 'F' WHERE SUM < 61;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000001' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL UPDATE OF 1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
  UPDATE TEMPTABLE SET GRADE = 'D' WHERE SUM BETWEEN 61 AND 70;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000001' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL UPDATE OF 1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
  UPDATE TEMPTABLE SET GRADE = 'C' WHERE SUM BETWEEN 71 AND 80;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD = 0 0 2 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000002' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL UPDATE OF 2 ROW(S)

```

Figure 122 (Part 5 of 23). Sample Output from Job DSNTEJ3P

```

1PAGE 1
***INPUT STATEMENT:
  UPDATE TEMPTABLE SET GRADE = 'B' WHERE SUM BETWEEN 81 AND 90;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD  = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD  = X'00000000' X'00000000' X'00000001' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL UPDATE OF 1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
  UPDATE TEMPTABLE SET GRADE = 'A' WHERE SUM > 90;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT416I SQLERRD  = 0 0 1 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD  = X'00000000' X'00000000' X'00000001' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
SUCCESSFUL UPDATE OF 1 ROW(S)
1PAGE 1
***INPUT STATEMENT:
  SELECT * FROM TEMPTABLE;

```

	NAME	MIDTERM	FINAL	SUM	GRADE
1	JOE	55	68	62	D
2	SUE	83	87	85	B
3	KEN	50	65	59	F
4	JAN	92	95	93	A
5	LES	67	75	71	C
6	SALLY	75	78	76	C

```

0SUCCESSFUL RETRIEVAL OF 6 ROW(S)

```

Figure 122 (Part 6 of 23). Sample Output from Job DSNTEJ3P

```

1PAGE 1
***INPUT STATEMENT:
ROLLBACK;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
ROLLBACK SUCCESSFUL
1PAGE 1
***INPUT STATEMENT:
SELECT NAME, BINDDATE
FROM SYSIBM.SYSPLAN
WHERE CREATOR = USER ;

```

	NAME	BINDDATE
1_	DSNTIA51	960605
2_	DSNESPRR	961010
3_	DSNESPCS	961010
4_	DSNEDCL	961010
5_	DSNHYCRD	961010
6_	DSNTEP51	961016
7_	DSNTIB51	961016
8_	DSN8BD51	961016
9_	DSN8BE51	961016
10_	DSN8BF51	961016
11_	DSN8BP51	961016
12_	DSN8BH51	961016
13_	DSN8SC51	961016
14_	DSN8HC51	961016

Figure 122 (Part 7 of 23). Sample Output from Job DSNTEJ3P

15_	DSN8SP51	961016
16_	DSN8EP	961016
17_	DSN8ED	961016
18_	MITSQL	960605
19_	DONSQL1	960605
20_	JUNKPL	961014
21_	JUNKP2	961007

0SUCCESSFUL RETRIEVAL OF 21 ROW(S)

1PAGE 1

***INPUT STATEMENT:

```
SELECT COLLID, NAME, VERSION, BINDTIME
FROM SYSIBM.SYSPACKAGE
WHERE OWNER = USER ;
```

	COLLID	NAME	VERSION
1_	DSNESP RR	DSNESM68	
2_	DSNESPCS	DSNESM68	
3_	DSNTEP2	DSNTEP2	
4_	DSN8HC51	DSN8HC3	
5_	DSN8STOR	DSN8EP1	
6_	DSN8STOR	DSN8ED1	
7_	DSN8ST1R	DSN8ED1	
8_	JUNKPKG	JUNKD	
9_	JUNKPKG	JUNKD2	
10_	DSNEDCL	DSNECP68	V5
11_	DSNHYCRD	DSNHYCRD	V5

| Figure 122 (Part 8 of 23). Sample Output from Job DSNTEJ3P

1PAGE 2

	BINDTIME
1_	1996-06-05-20.23.26.358559
2_	1996-06-05-20.23.26.022395
3_	1996-10-16-15.35.44.991448
4_	1996-10-16-16.11.19.801602
5_	1996-10-16-12.34.11.347714
6_	1996-10-16-12.36.25.452740
7_	1996-10-02-17.44.24.302450
8_	1996-10-14-17.47.55.901201
9_	1996-10-07-19.16.52.858386
10_	1996-10-10-07.07.46.539201
11_	1996-10-10-07.07.47.304488

0SUCCESSFUL RETRIEVAL OF 11 ROW(S)

1PAGE 1

***INPUT STATEMENT:

```
SELECT NAME, CREATOR, BINDDATE, VALID, OPERATIVE
FROM SYSIBM.SYSPLAN
WHERE OPERATIVE = 'N' OR VALID = 'N';
```

0SUCCESSFUL RETRIEVAL OF 0 ROW(S)

Figure 122 (Part 9 of 23). Sample Output from Job DSNTEJ3P

1PAGE 1

***INPUT STATEMENT:

```
SELECT COLLID, NAME, VERSION, BINDTIME, VALID
FROM SYSIBM.SYSPACKAGE
WHERE OPERATIVE = 'N' OR VALID = 'N';
```

	COLLID	NAME	VERSION
1_	JUNKPKG	JUNKD	
2_	JUNKPKG	JUNKD2	

1PAGE 2

	BINDTIME	VALID
1_	1996-10-14-17.47.55.901201	N
2_	1996-10-07-19.16.52.858386	N

0SUCCESSFUL RETRIEVAL OF 2 ROW(S)

Figure 122 (Part 10 of 23). Sample Output from Job DSNTEJ3P

1PAGE 1

***INPUT STATEMENT:

```
SELECT DNAME, BTYPE, BCREATOR, BNAME
FROM SYSIBM.SYSPLANDEP
WHERE BCREATOR = USER
ORDER BY DNAME, BTYPE, BCREATOR, BNAME;
```

	DNAME	BTYPE	BCREATOR	BNAME
1_	DSN8BD51	S	SYSADM	VEMPLP
2_	DSN8BD51	S	SYSADM	VPHONE
3_	DSN8BE51	S	SYSADM	VEMPLP
4_	DSN8BE51	S	SYSADM	VPHONE
5_	DSN8BF51	S	SYSADM	VEMPLP
6_	DSN8BF51	S	SYSADM	VPHONE
7_	DSN8BH51	S	SYSADM	VEMPLP
8_	DSN8BH51	S	SYSADM	VPHONE
9_	DSN8BP51	S	SYSADM	VEMPLP
10_	DSN8BP51	S	SYSADM	VPHONE
11_	DSN8SC51	S	SYSADM	VEMPLP
12_	DSN8SC51	S	SYSADM	VPHONE
13_	DSN8SP51	S	SYSADM	VEMPLP
14_	DSN8SP51	S	SYSADM	VPHONE

0SUCCESSFUL RETRIEVAL OF 14 ROW(S)

1PAGE 1

***INPUT STATEMENT:

```
SELECT DCOLLID, DNAME, BTYPE, BQUALIFIER, BNAME
FROM SYSIBM.SYSPACKDEP
WHERE BQUALIFIER = USER
ORDER BY DCOLLID, DNAME, BTYPE, BQUALIFIER, BNAME;
```

	DCOLLID	DNAME	BTYPE	BQUALIFIER	BNAME
1_	DSN8HC51	DSN8HC3	S	SYSADM	VEMP
2_	DSN8HC51	DSN8HC3	S	SYSADM	VHDEPT

0SUCCESSFUL RETRIEVAL OF 2 ROW(S)

1PAGE 1

***INPUT STATEMENT:

```
SELECT NAME, TYPE
FROM SYSIBM.SYSTABLES
WHERE CREATOR = USER
ORDER BY NAME, TYPE;
```

	NAME	TYPE
1_	DSNRLST01	T
2_	GLOTZ	G
3_	PLAN_TABLE	T

0SUCCESSFUL RETRIEVAL OF 3 ROW(S)

Figure 122 (Part 11 of 23). Sample Output from Job DSNTEJ3P

1PAGE 1

***INPUT STATEMENT:

SELECT *
FROM SYSIBM.SYSTABLES
WHERE NAME = 'DEPT';

	NAME	CREATOR	TYPE	DBNAME	TSNAME	DBID	OBID	COLCOUNT	EDPROC	VALPROC
1_	DEPT	DSN8510	T	DSN8D51A	DSN8S51D	259	11	5		
2_	DEPT	USRT027	T	DB2	TS1	262	10	4		
3_	DEPT	ADMF001	T	RUNNING	TABRUN	264	5	4		
4_	DEPT	USRT029	T	RDB	RTS1	265	21	3		
5_	DEPT	USRT039	T	RDB	RTS8	265	127	3		

1PAGE 2

	CLUSTERTYPE	CLUSTERRID	CARD	NPAGES	PCTPAGES	IBMREQD
1_		0	14	1	8	C
2_		0	-1	-1	-1	N
3_		0	-1	-1	-1	N
4_		0	-1	-1	-1	N
5_		0	-1	-1	-1	N

1PAGE 3

	REMARKS
1_	
2_	
3_	
4_	
5_	

1PAGE 4

	PARENTS	CHILDREN	KEYCOLUMNS	RECLENGTH	STATUS	KEYOBID	LABEL	CHECKFLAG
1_	2	3	1	76	X	12		
2_	0	0	0	62		0		
3_	0	0	0	63		0		
4_	0	0	0	54		0		
5_	0	0	0	54		0		

1PAGE 5

	CHECKRID	AUDITING	CREATEDBY	LOCATION	TBCREATOR	TBNAME	CREATEDTS
1_			SYSADM				1996-06-05-18.12.03.562822
2_			USRT027				1996-06-05-19.35.17.482711
3_			ADMF001				1996-06-05-19.40.34.317220
4_			ADMF001				1996-06-05-19.41.25.726052
5_			ADMF001				1996-06-05-19.42.27.645219

Figure 122 (Part 12 of 23). Sample Output from Job DSNTEJ3P

1PAGE 6

	ALTEREDTS	DATACAPTURE	RBA1	RBA2	PCTROWCOMP	STATSTIME	CHECKS
1	1996-06-05-18.12.27.239548		0	1996-06-05-18.13.36.060817	0
2	1996-06-05-19.35.17.482711		-1	0001-01-01-00.00.00.000000	0
3	1996-06-05-19.40.34.317220		-1	0001-01-01-00.00.00.000000	0
4	1996-06-05-19.41.25.726052		-1	0001-01-01-00.00.00.000000	0
5	1996-06-05-19.42.27.645219		-1	0001-01-01-00.00.00.000000	0

1PAGE 7

	CARDF	CHECKRID5B	ENCODING_SCHEME
1	1.4000000000000000E+01		E
2	-1.0000000000000000E+00		E
3	-1.0000000000000000E+00		E
4	-1.0000000000000000E+00		E
5	-1.0000000000000000E+00		E

0SUCCESSFUL RETRIEVAL OF 5 ROW(S)

1PAGE 1

***INPUT STATEMENT:

```

SELECT NAME, COLTYPE, LENGTH, SCALE, NULLS, REMARKS, COLNO
FROM SYSIBM.SYSCOLUMNS
WHERE TBNAME= 'DEPT' AND TBCREATOR = 'DSN8510'
ORDER BY COLNO;

```

	NAME	COLTYPE	LENGTH	SCALE	NULLS
1	DEPTNO	CHAR	3	0	N
2	DEPTNAME	VARCHAR	36	0	N
3	MGRNO	CHAR	6	0	Y
4	ADMUDEPT	CHAR	3	0	N
5	LOCATION	CHAR	16	0	Y

1PAGE 2

	REMARKS
1	
2	
3	
4	
5	

1PAGE 3

	COLNO
1	1
2	2
3	3
4	4
5	5

0SUCCESSFUL RETRIEVAL OF 5 ROW(S)

Figure 122 (Part 13 of 23). Sample Output from Job DSNTEJ3P

1PAGE 1

***INPUT STATEMENT:

SELECT * FROM SYSIBM.SYSPLANAUTH WHERE GRANTEE = USER;

	GRANTOR	GRANTEE	NAME	TIMESTAMP	DATEGRANTED	TIMEGRANTED	GRANTEEType	AUTHHOWGOT	BINDAUTH
1_	SYSADM	SYSADM	DSNTIA51	CD08K74HVIY8	960605	20213254			G
2_	SYSADM	SYSADM	DSNESPRR	CD08LNYGDJBD	960605	20232640			G
3_	SYSADM	SYSADM	DSNESPCS	CD08LNWWNP50	960605	20232608			G
4_	SYSADM	SYSADM	DSNEDCL	CD08LYAGI8PP	960605	20244046			G
5_	SYSADM	SYSADM	DSNHYCRD	CD07JI25JFNU	960605	17475221			G
6_	ADMFO01	SYSADM	DSNTEP3	CEFNH3YJFIUF	960904	10481436			G
7_	SYSADM	SYSADM	DSNTEP51	CD07P5U9TPZJ	960605	18153673			G
8_	SYSADM	SYSADM	DSNTIB51	CD07QPT1153D	960605	18180009			G
9_	SYSADM	SYSADM	DSN8BD51	CD07Q8ZB1UB2	960605	18201043			G
10_	SYSADM	SYSADM	DSN8BF51	CD07RWP4DC9B	960605	18230088			G
11_	SYSADM	SYSADM	DSN8BP51	CD07SUNV1KCZ	960605	18265739			G
12_	SYSADM	SYSADM	DSN8BH51	CD07UYRU3VDI	960605	18354957			G
13_	SYSADM	SYSADM	DSN8SC51	CD07VBRXT7N2	960605	18371576			G
14_	SYSADM	SYSADM	DSN8HC51	CD07VCTINLE4	960605	18372326			G
15_	SYSADM	SYSADM	DSN8SP51	CD07VTVURJUQ	960605	18392582			G
16_	SYSADM	SYSADM	DSN8EP	CEMZTWV5GWXQ	960930	15353847			G
17_	SYSADM	SYSADM	DSN8ED	CENAMH6RJKA9	961001	15303734			G
18_	SYSADM	SYSADM	MITSQL1	CD08LD5WC150	960605	20221591			G
19_	SYSADM	SYSADM	DONSQL1	CD08MHJ4HN3D	960605	20265170			G
20_	SYSADM	SYSADM	JUNKPL	CEN1NLT8X2YJ	961004	07070111			G
21_	SYSADM	SYSADM	JUNKP2	CEN2FFKVXHLY	961004	08592190			G
22_	SYSADM	SYSADM	DSN8BE51	CERH2MW7IAER	961016	15451044			G

1PAGE 2

	EXECUTEAUTH	IBMREQD	GRANTEDTS
1_	G	N	1996-06-05-20.21.32.544451
2_	G	N	1996-06-05-20.23.26.401360
3_	G	N	1996-06-05-20.23.26.086529
4_	G	N	1996-06-05-20.24.40.466364
5_	G	N	1996-06-05-17.47.52.215894
6_		N	1996-09-04-10.48.14.366360
7_	G	N	1996-06-05-18.15.36.733144
8_	G	N	1996-06-05-18.18.00.096832
9_	G	N	1996-06-05-18.20.10.434268
10_	G	N	1996-06-05-18.23.00.880232
11_	G	N	1996-06-05-18.26.57.390825
12_	G	N	1996-06-05-18.35.49.579662
13_	G	N	1996-06-05-18.37.15.764495
14_	G	N	1996-06-05-18.37.23.266514
15_	G	N	1996-06-05-18.39.25.820172
16_	G	N	1996-09-30-15.35.38.473263
17_	G	N	1996-10-01-15.30.37.343062
18_	G	N	1996-06-05-20.22.15.918797
19_	G	N	1996-06-05-20.26.51.700448
20_	G	N	1996-10-04-07.07.01.119952
21_	G	N	1996-10-04-08.59.21.903908
22_	G	N	1996-10-16-15.45.10.440560

0SUCCESSFUL RETRIEVAL OF 22 ROW(S)

Figure 122 (Part 14 of 23). Sample Output from Job DSNTJ3P

1PAGE 1

***INPUT STATEMENT:

SELECT * FROM SYSIBM.SYSPACKAUTH WHERE GRANTEE = USER;

	GRANTOR	GRANTEE	LOCATION	COLLID	NAME	CONTOKEN	TIMESTAMP
1_	SYSADM	SYSADM		DSNESPRR	DSNESM68		1996-06-05-20.23.26.358559
2_	SYSADM	SYSADM		DSNESPCS	DSNESM68		1996-06-05-20.23.26.022395
3_	SYSADM	SYSADM		DSNTEP2	DSNTEP2		1996-06-05-18.15.36.290592
4_	SYSADM	SYSADM		DSN8HC51	DSN8HC3		1996-06-05-18.37.22.213551
5_	ADMFO01	SYSADM		DSNTEP3	DSNTEP3		1996-09-04-10.48.13.473781
6_	SYSADM	SYSADM		DSN8STOR	DSN8EP1		1996-09-30-15.35.25.122831
7_	SYSADM	SYSADM		DSN8STOR	DSN8ED1		1996-10-01-15.30.12.706216
8_	SYSADM	SYSADM		DSN8ST1R	DSN8ED1		1996-10-02-17.38.27.585188
9_	SYSADM	SYSADM		JUNKPKG	JUNKD		1996-10-04-07.06.56.561753
10_	SYSADM	SYSADM		JUNKPKG	JUNKD2		1996-10-04-08.59.16.910638
11_	SYSADM	SYSADM		DSNEDCL	DSNECP68		1996-10-10-07.07.46.539201
12_	SYSADM	SYSADM		DSNHYCRD	DSNHYCRD		1996-10-10-07.07.47.304488

1PAGE 2

	GRANTEETYPE	AUTHHOWGOT	BINDAUTH	COPYAUTH	EXECUTEAUTH	IBMREQD
1_			G	G	G	N
2_			G	G	G	N
3_			G	G	G	N
4_			G	G	G	N
5_			G	G	G	N
6_			G	G	G	N
7_			G	G	G	N
8_			G	G	G	N
9_			G	G	G	N
10_			G	G	G	N
11_			G	G	G	N
12_			G	G	G	N

0SUCCESSFUL RETRIEVAL OF 12 ROW(S)

1PAGE 1

***INPUT STATEMENT:

SELECT * FROM SYSIBM.SYSUSERAUTH WHERE GRANTEE = USER;

0SUCCESSFUL RETRIEVAL OF 0 ROW(S)

Figure 122 (Part 15 of 23). Sample Output from Job DSNTEJ3P

1PAGE 1

***INPUT STATEMENT:

SELECT * FROM SYSIBM.SYSDBAUTH WHERE GRANTEE = USER;

	GRANTOR	GRANTEE	NAME	TIMESTAMP	DATEGRANTED	TIMEGRANTED	GRANTEETYPE	AUTHHOWGOT	CREATETABAUTH
1	SYSADM	SYSADM	DSNDB07	CD07IRN7BZ78	960605	17444265	S	G	G
2	SYSADM	SYSADM	DSNRLST	CD07JI74B6QT	960605	17475323	S	G	G
3	SYSADM	SYSADM	DSNDDF	CD07JJVP8AG	960605	17475807	S	G	G
4	SYSADM	SYSADM	DSNRGFDB	CD07JNBE14R7	960605	17482263	S	G	G
5	SYSADM	SYSADM	DSN8D51A	CD0709IA00RW	960605	18115146	S	G	G
6	SYSADM	SYSADM	DSN8D51P	CD0709I2DRKO	960605	18115162	S	G	G
7	SYSADM	SYSADM	WALKING	CD08BANRJUVT	960605	19395760	S	G	G
8	SYSADM	SYSADM	DSN8D51U	CERHZ2SRU8DB	961016	15383459	S	G	G

1PAGE 2

	CREATETSAUTH	DBADMAUTH	DBCTRLAUTH	DBMAINTAUTH	DISPLAYDBAUTH	DROPAUTH	IMAGCOPYAUTH	LOADAUTH	REORGAUTH
1	G	G	G	G	G	G	G	G	G
2	G	G	G	G	G	G	G	G	G
3	G	G	G	G	G	G	G	G	G
4	G	G	G	G	G	G	G	G	G
5	G	G	G	G	G	G	G	G	G
6	G	G	G	G	G	G	G	G	G
7	G	G	G	G	G	G	G	G	G
8	G	G	G	G	G	G	G	G	G

1PAGE 3

	RECOVERDBAUTH	REPAIRAUTH	STARTDBAUTH	STATSAUTH	STOPAUTH	IBMREQD	GRANTEDTS
1	G	G	G	G	G	N	1996-06-05-17.44.42.655234
2	G	G	G	G	G	N	1996-06-05-17.47.53.234635
3	G	G	G	G	G	N	1996-06-05-17.47.58.079685
4	G	G	G	G	G	N	1996-06-05-17.48.22.635822
5	G	G	G	G	G	N	1996-06-05-18.11.51.468558
6	G	G	G	G	G	N	1996-06-05-18.11.51.624998
7	G	G	G	G	G	N	1996-06-05-19.39.57.606890
8	G	G	G	G	G	N	1996-10-16-15.38.34.594266

0SUCCESSFUL RETRIEVAL OF 8 ROW(S)

1PAGE 1

***INPUT STATEMENT:

SELECT * FROM SYSIBM.SYSTABAUTH WHERE GRANTEE = USER;

	GRANTOR	GRANTEE	GRANTEETYPE	DBNAME	SCREATOR	STNAME	TCREATOR	TTNAME
1	SYSADM	SYSADM			SYSADM	GLOTZ	SYSADM	GLOTZ
2	SYSADM	SYSADM			SYSADM	DSNRLST01	SYSADM	DSNRLST01
3	SYSADM	SYSADM			SYSADM	PLAN_TABLE	SYSADM	PLAN_TABLE

Figure 122 (Part 16 of 23). Sample Output from Job DSNTEJ3P

1PAGE 2

	AUTHHOWGOT	TIMESTAMP	DATEGRANTED	TIMEGRANTED	UPDATECOLS	ALTERAUTH	DELETAUTH	INDEXAUTH	INSERTAUTH
1_		CEQX9F7KIR7Z	961014	17480999		G	G	G	G
2_	S	CD07JJGXS35	960605	17475494		G	G	G	G
3_	S	CD07VWJC3WA8	960605	18394479		G	G	G	G

1PAGE 3

	SELECTAUTH	UPDATEAUTH	IBMREQD	GRANTEELLOCATION	LOCATION	COLLID	CONTOKEN	CAPTUREAUTH
1_	G	G	N					
2_	G	G	N					
3_	G	G	N					

1PAGE 4

	REFERENCESAUTH	REFCOLS	GRANTEDTS
1_	G		1996-10-14-17.48.09.999246
2_	G		1996-06-05-17.47.54.949910
3_	G		1996-06-05-18.39.44.796773

0SUCCESSFUL RETRIEVAL OF 3 ROW(S)

1PAGE 1

***INPUT STATEMENT:

SELECT * FROM SYSIBM.SYSCOLAUTH WHERE GRANTEE = USER;

0SUCCESSFUL RETRIEVAL OF 0 ROW(S)

1PAGE 1

***INPUT STATEMENT:

SELECT * FROM SYSIBM.SYSRESAUTH WHERE GRANTEE = USER;

	GRANTOR	GRANTEE	QUALIFIER	NAME	GRANTEETYPE	AUTHHOWGOT	OBTYPE	TIMESTAMP	DATEGRANTED
1_	SYSADM	SYSADM		SYSDEFLT		S	S	CD07JI4U4BQX	960605
2_	SYSADM	SYSADM		DSN8G510		S	S	CD0709ECVPQV	960605
3_	SYSADM	SYSADM		DSN8G51U		S	S	CERHZ2QVR2HW	961016
4_	SYSADM	SYSADM		SG1		S	S	CD0793KX65YT	960605
5_	SYSADM	SYSADM		WALKSG		S	S	CD08BAMS9VEU	960605
6_	SYSADM	SYSADM		RSG1		S	S	CD08BKOFAXJW	960605

1PAGE 2

	TIMEGRANTED	USEAUTH	IBMREQD	GRANTEDTS
1_	17475257	G	N	1996-06-05-17.47.52.570934
2_	18115066	G	N	1996-06-05-18.11.50.660804
3_	15383420	G	N	1996-10-16-15.38.34.206855
4_	19345543	G	N	1996-06-05-19.34.55.439849
5_	19395741	G	N	1996-06-05-19.39.57.411778
6_	19410954	G	N	1996-06-05-19.41.09.547468

0SUCCESSFUL RETRIEVAL OF 6 ROW(S)

Figure 122 (Part 17 of 23). Sample Output from Job DSNTEJ3P

1PAGE 1

***INPUT STATEMENT:

```
SELECT TCREATOR, TTNAME, STNAME, GRANTOR
FROM SYSIBM.SYSTABAUTH
WHERE GRANTEE = USER;
```

	TCREATOR	TTNAME	STNAME	GRANTOR
1_	SYSADM	GLOTZ	GLOTZ	SYSADM
2_	SYSADM	DSNRLST01	DSNRLST01	SYSADM
3_	SYSADM	PLAN_TABLE	PLAN_TABLE	SYSADM

0SUCCESSFUL RETRIEVAL OF 3 ROW(S)

1PAGE 1

***INPUT STATEMENT:

```
SELECT BNAME, BTYPE, GRANTOR, NAME
FROM SYSIBM.SYSPLANAUTH, SYSIBM.SYSPLANDEP
WHERE GRANTEE = USER
AND NAME = DNAME
AND EXECUTEAUTH = ' '
AND ( BTYPE = 'T' OR BTYPE = 'V');
```

	BNAME	BTYPE	GRANTOR	NAME
1_	VPHONE	V	SYSADM	DSN8BD51
2_	EMP	T	SYSADM	DSN8BD51
3_	VEMPLP	V	SYSADM	DSN8BD51
4_	DEPT	T	SYSADM	DSN8BD51
5_	EMP	T	SYSADM	DSN8BE51
6_	VPHONE	V	SYSADM	DSN8BE51
7_	VEMPLP	V	SYSADM	DSN8BE51
8_	DEPT	T	SYSADM	DSN8BE51
9_	VEMPLP	V	SYSADM	DSN8BF51
10_	EMP	T	SYSADM	DSN8BF51
11_	VPHONE	V	SYSADM	DSN8BF51
12_	DEPT	T	SYSADM	DSN8BF51
13_	VPHONE	V	SYSADM	DSN8BH51
14_	VEMPLP	V	SYSADM	DSN8BH51
15_	DEPT	T	SYSADM	DSN8BH51
16_	EMP	T	SYSADM	DSN8BH51
17_	VPHONE	V	SYSADM	DSN8BP51
18_	VEMPLP	V	SYSADM	DSN8BP51
19_	DEPT	T	SYSADM	DSN8BP51
20_	EMP	T	SYSADM	DSN8BP51
21_	EMP	T	SYSADM	DSN8SC51
22_	DEPT	T	SYSADM	DSN8SC51
23_	VPHONE	V	SYSADM	DSN8SC51
24_	VEMPLP	V	SYSADM	DSN8SC51
25_	VPHONE	V	SYSADM	DSN8SP51
26_	DEPT	T	SYSADM	DSN8SP51
27_	EMP	T	SYSADM	DSN8SP51
28_	VEMPLP	V	SYSADM	DSN8SP51

0SUCCESSFUL RETRIEVAL OF 28 ROW(S)

Figure 122 (Part 18 of 23). Sample Output from Job DSNTEJ3P

1PAGE 1

***INPUT STATEMENT:

```
SELECT DCOLLID, BNAME, BTYPE, BQUALIFIER, BNAME
FROM SYSIBM.SYSPACKAUTH, SYSIBM.SYSPACKDEP
WHERE GRANTEE = USER
AND COLLID = DCOLLID
AND NAME = DNAME
AND EXECUTEAUTH ^= ' '
AND ( BTYPE = 'T' OR BTYPE = 'V');
```

	DCOLLID	BNAME	BTYPE	BQUALIFIER	BNAME
1_	DSNHYCRD	SYSTABLES	T	SYSIBM	SYSTABLES
2_	DSNHYCRD	SYSCOLUMNS	T	SYSIBM	SYSCOLUMNS
3_	DSNHYCRD	SYSVIEWS	T	SYSIBM	SYSVIEWS
4_	DSN8HC51	VHDEPT	V	DSN8510	VHDEPT
5_	DSN8HC51	DEPT	T	DSN8510	DEPT
6_	DSN8HC51	VEMP	V	DSN8510	VEMP
7_	DSN8HC51	EMP	T	DSN8510	EMP

0SUCCESSFUL RETRIEVAL OF 7 ROW(S)

1PAGE 1

***INPUT STATEMENT:

```
SELECT NAME, CREATOR, TYPE, DBNAME, TSNAME
FROM SYSIBM.SYSTABLES
WHERE DBNAME IN
(SELECT NAME FROM SYSIBM.SYSDBAUTH
WHERE GRANTEE = USER
AND DBDMAUTH ^= ' ');
```

	NAME	CREATOR	TYPE	DBNAME	TSNAME
1_	DSNRLST01	SYSADM	T	DSNRLST	DSNRLS01
2_	SYSLUNAMES	SYSIBM	T	DSNDDF	SYSDDF
3_	SYSLULIST	SYSIBM	T	DSNDDF	SYSDDF
4_	SYSLOCATIONS	SYSIBM	T	DSNDDF	SYSDDF
5_	SYSLMODES	SYSIBM	T	DSNDDF	SYSDDF
6_	SYSMODESELECT	SYSIBM	T	DSNDDF	SYSDDF
7_	SYSUSERNAMES	SYSIBM	T	DSNDDF	SYSDDF
8_	DSN_REGISTER_APPL	DSNRGCOL	T	DSNRGFDB	DSNRGFTS
9_	DSN_REGISTER_OBJT	DSNRGCOL	T	DSNRGFDB	DSNRGFTS
10_	DEPT	DSN8510	T	DSN8D51A	DSN8S51D
11_	VDEPT	DSN8510	V	DSN8D51A	DSN8S51D
12_	VHDEPT	DSN8510	V	DSN8D51A	DSN8S51D
13_	EMP	DSN8510	T	DSN8D51A	DSN8S51E
14_	VEMP	DSN8510	V	DSN8D51A	DSN8S51E
15_	TCONA	DSN8510	T	DSN8D51P	DSN8S51C
16_	VDSPTXT	DSN8510	V	DSN8D51P	DSN8S51C
17_	TOPTVAL	DSN8510	T	DSN8D51P	DSN8S51C
18_	EDEPT	DSN8510	T	DSN8D51A	DSN8S51R
19_	EEMP	DSN8510	T	DSN8D51A	DSN8S51R
20_	EPROJ	DSN8510	T	DSN8D51A	DSN8S51R
21_	PROJ	DSN8510	T	DSN8D51A	DSN8S51P
22_	VPROJACT	DSN8510	V	DSN8D51A	DSN8S51P
23_	VEMPPROJACT	DSN8510	V	DSN8D51A	DSN8S51P
24_	ACT	DSN8510	T	DSN8D51A	DSN8S51P
25_	PARTS	DSN8510	T	DSN8D51A	DSN8S51S

| Figure 122 (Part 19 of 23). Sample Output from Job DSNTEJ3P

26	VACT	DSN8510	V	DSN8D51A	DSN8S51P
27	PROJACT	DSN8510	T	DSN8D51A	DSN8S51P
28	VPROJ	DSN8510	V	DSN8D51A	DSN8S51P
29	EMPPROJACT	DSN8510	T	DSN8D51A	DSN8S51P
30	VOPTVAL	DSN8510	V	DSN8D51P	DSN8S51C
31	TDSPTXT	DSN8510	T	DSN8D51P	DSN8S51C
32	VCONA	DSN8510	V	DSN8D51P	DSN8S51C
33	EACT	DSN8510	T	DSN8D51A	DSN8S51R
34	EPROJACT	DSN8510	T	DSN8D51A	DSN8S51R
35	EEPA	DSN8510	T	DSN8D51A	DSN8S51R
36	VDEPMG1	DSN8510	V	DSN8D51A	DSN8S51D
37	VASTRDE2	DSN8510	V	DSN8D51A	DSN8S51E
38	VEMPDPT1	DSN8510	V	DSN8D51A	DSN8S51E
39	VPHONE	DSN8510	V	DSN8D51A	DSN8S51E
40	VSTAFAC1	DSN8510	V	DSN8D51A	DSN8S51P
41	VPROJRE1	DSN8510	V	DSN8D51A	DSN8S51P

1PAGE 2

	NAME	CREATOR	TYPE	DBNAME	TSNAME
42	VFORPLA	DSN8510	V	DSN8D51A	DSN8S51P
43	VSTAFAC2	DSN8510	V	DSN8D51A	DSN8S51P
44	VEMPLP	DSN8510	V	DSN8D51A	DSN8S51E
45	NEWPHONE	DSN8510	T	DSN8D51U	NEWPHONE
46	PLAN_TABLE	SYSADM	T	DSN8D51A	DSN8S51X
47	SORTTAB1	USRT027	T	WALKING	WALK
48	SORTTAB3	USRT027	T	WALKING	WALK
49	SORTTAB2	USRT027	T	WALKING	WALK2
50	NEWDEPT	DSN8510	T	DSN8D51U	NEWDEPT

0SUCCESSFUL RETRIEVAL OF 50 ROW(S)

1

1PAGE 1

***INPUT STATEMENT:

DROP TABLESPACE DSN8D51A.DSN8S51X;

SQLERROR ON DROP COMMAND, EXECUTE FUNCTION

RESULT OF SQL STATEMENT:

DSNT408I SQLCODE = -204, ERROR: DSN8D51A.DSN8S51X IS AN UNDEFINED NAME

DSNT418I SQLSTATE = 42704 SQLSTATE RETURN CODE

DSNT415I SQLERRP = DSNXISUB SQL PROCEDURE DETECTING ERROR

DSNT416I SQLERRD = 95 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION

DSNT416I SQLERRD = X'0000005F' X'00000000' X'00000000' X'FFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION

1PAGE 1

***INPUT STATEMENT:

COMMIT;

RESULT OF SQL STATEMENT:

DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION

COMMIT SUCCESSFUL

1PAGE 1

***INPUT STATEMENT:

DROP TABLESPACE DSN8D51A.DSN8S51X;

RESULT OF SQL STATEMENT:

DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION

DROP SUCCESSFUL

1PAGE 1

***INPUT STATEMENT:

COMMIT;

RESULT OF SQL STATEMENT:

DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION

COMMIT SUCCESSFUL

Figure 122 (Part 20 of 23). Sample Output from Job DSNTEJ3P


```

1PAGE 1
***INPUT STATEMENT:
CREATE TABLESPACE DSN8S51X
  IN DSN8D51A
  USING STOGROUP DSN8G510;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
CREATE SUCCESSFUL
1PAGE 1
***INPUT STATEMENT:
CREATE TABLE PLAN_TABLE
( QUERYNO          INTEGER          NOT NULL,
  QBLOCKNO        SMALLINT        NOT NULL,
  APPLNAME        CHAR(8)          NOT NULL,
  PROGNAME        CHAR(8)          NOT NULL,
  PLANNO          SMALLINT        NOT NULL,
  METHOD           SMALLINT        NOT NULL,
  CREATOR         CHAR(8)          NOT NULL,
  TNAME           CHAR(18)         NOT NULL,
  TABNO           SMALLINT        NOT NULL,
  ACCESTYPE       CHAR(2)          NOT NULL,
  MATCHCOLS       SMALLINT        NOT NULL,
  ACCESSCREATOR   CHAR(8)          NOT NULL,
  ACCESSNAME      CHAR(18)         NOT NULL,
  INDEXONLY       CHAR(1)          NOT NULL,
  SORTN_UNIQ      CHAR(1)          NOT NULL,
  SORTN_JOIN      CHAR(1)          NOT NULL,
  SORTN_ORDERBY   CHAR(1)          NOT NULL,
  SORTN_GROUPBY   CHAR(1)          NOT NULL,
  SORTC_UNIQ      CHAR(1)          NOT NULL,
  SORTC_JOIN      CHAR(1)          NOT NULL,
  SORTC_ORDERBY   CHAR(1)          NOT NULL,
  SORTC_GROUPBY   CHAR(1)          NOT NULL,
  TSLOCKMODE      CHAR(3)          NOT NULL,
  TIMESTAMP       CHAR(16)         NOT NULL,
  REMARKS         VARCHAR(254)     NOT NULL,
  PREFETCH        CHAR(1)          NOT NULL,
  COLUMN_FN_EVAL  CHAR(1)          NOT NULL,
  MIXOPSEQ        SMALLINT        NOT NULL,
  VERSION         VARCHAR(64)      NOT NULL,
  COLLID          CHAR(18)         NOT NULL,
  ACCESS_DEGREE   SMALLINT,
  ACCESS_PGROUP_ID SMALLINT,
  JOIN_DEGREE     SMALLINT,
  JOIN_PGROUP_ID  SMALLINT,
  SORTC_PGROUP_ID SMALLINT,
  SORTN_PGROUP_ID SMALLINT,
  PARALLELISM_MODE CHAR(1),
  MERGE_JOIN_COLS SMALLINT,
  CORRELATION_NAME CHAR(18),
  PAGE_RANGE      CHAR(1)          NOT NULL,
  JOIN_TYPE       CHAR(1)          NOT NULL,
  GROUP_MEMBER    CHAR(8)          NOT NULL,
  IBM_SERVICE_DATA VARCHAR(254)    NOT NULL,
  WHEN_OPTIMIZE   CHAR(1)          NOT NULL WITH DEFAULT,
  QBLOCK_TYPE     CHAR(6)          NOT NULL,
  BIND_TIME       TIMESTAMP       NOT NULL
)
  IN DSN8D51A.DSN8S51X
  CCSID EBCDIC;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
CREATE SUCCESSFUL

```

Figure 122 (Part 21 of 23). Sample Output from Job DSNTEJ3P

```

1PAGE 1
***INPUT STATEMENT:
  EXPLAIN PLAN SET QUERYNO = 9 FOR
  SELECT DISTINCT
    DEPTNO, DEPTNAME, MGRNO,
    FIRSTNAME, MIDINIT, LASTNAME
  FROM VDEPMG1
  WHERE DEPTNO LIKE 'D_1'
    AND DEPTNAME LIKE '%'
    AND MGRNO LIKE '%'
    AND LASTNAME LIKE '%'
    AND '%' = '%'
    AND '%' = '%'
    AND DEPTNO > 'D01';
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
EXPLAIN SUCCESSFUL
1PAGE 1
***INPUT STATEMENT:
  SELECT * FROM PLAN_TABLE WHERE QUERYNO = 9 ;
SQLWARNING ON SELECT COMMAND, FETCH FUNCTION
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT418I SQLSTATE = 01004 SQLSTATE RETURN CODE
DSNT416I SQLERRD = 0 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000000' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
DSNT417I SQLWARN0-5 = W,W,,, SQL WARNINGS
DSNT417I SQLWARN6-A = ,,,, SQL WARNINGS
SELECT SUCCESSFUL
SQLWARNING ON SELECT COMMAND, FETCH FUNCTION
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT418I SQLSTATE = 01004 SQLSTATE RETURN CODE
DSNT416I SQLERRD = 0 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000000' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
DSNT417I SQLWARN0-5 = W,W,,, SQL WARNINGS
DSNT417I SQLWARN6-A = ,,,, SQL WARNINGS
SELECT SUCCESSFUL
SQLWARNING ON SELECT COMMAND, FETCH FUNCTION
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
DSNT418I SQLSTATE = 01004 SQLSTATE RETURN CODE
DSNT416I SQLERRD = 0 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000000' X'00000000' X'00000000' X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
DSNT417I SQLWARN0-5 = W,W,,, SQL WARNINGS
DSNT417I SQLWARN6-A = ,,,, SQL WARNINGS
SELECT SUCCESSFUL
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| QUERYNO | QBLOCKNO | APPLNAME | PROGNAME | PLANNO | METHOD | CREATOR | TNAME | TABNO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1_| 9 | 1 | | DSNTEP2 | 1 | 0 | DSN8510 | DEPT | 1 |
2_| 9 | 1 | | DSNTEP2 | 2 | 1 | DSN8510 | EMP | 2 |
3_| 9 | 1 | | DSNTEP2 | 3 | 3 | | | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1PAGE 2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ACCESSTYPE | MATCHCOLS | ACCESSCREATOR | ACCESSNAME | INDEXONLY | SORTN_UNIQ | SORTN_JOIN | SORTN_ORDERBY |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1_| R | 0 | | | N | N | N | N |
2_| I | 1 | DSN8510 | XEMP1 | N | N | N | N |
3_| | 0 | | | N | N | N | N |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 122 (Part 22 of 23). Sample Output from Job DSNTEJ3P

1PAGE 3

	SORTN_GROUPBY	SORTC_UNIQ	SORTC_JOIN	SORTC_ORDERBY	SORTC_GROUPBY	TSLOCKMODE	TIMESTAMP
1_	N	N	N	N	N	IS	1996101616214289
2_	N	N	N	N	N	IS	1996101616214289
3_	N	Y	N	N	N		1996101616214289

1PAGE 4

	REMARKS
1_	
2_	
3_	

1PAGE 5

	PREFETCH	COLUMN_FN_EVAL	MIXOPSEQ	VERSION
1_	S		0	
2_			0	
3_			0	

1PAGE 6

	COLLID	ACCESS_DEGREE	ACCESS_PGROUP_ID	JOIN_DEGREE	JOIN_PGROUP_ID	SORTC_PGROUP_ID	SORTN_PGROUP_ID
1_	DSNTEP2	?	?	?	?	?	?
2_	DSNTEP2	?	?	?	?	?	?
3_	DSNTEP2	?	?	?	?	?	?

1PAGE 7

	PARALLELISM_MODE	MERGE_JOIN_COLS	CORRELATION_NAME	PAGE_RANGE	JOIN_TYPE	GROUP_MEMBER
1_	?	?				
2_	?	?			L	
3_	?	?				

1PAGE 8

	IBM_SERVICE_DATA
1_	...
2_	...
3_	...

1PAGE 9

	WHEN_OPTIMIZE	QBLOCK_TYPE	BIND_TIME
1_		SELECT	1996-10-16-16.21.42.890000
2_		SELECT	1996-10-16-16.21.42.890000
3_		SELECT	1996-10-16-16.21.42.890000

0SUCCESSFUL RETRIEVAL OF 3 ROW(S)

Figure 122 (Part 23 of 23). Sample Output from Job DSNTEJ3P

Phases 4 and 5

The sample output from Phases 4 and 5 has not been included in this appendix as the output varies from customer to customer. For all other jobs, see the return code tables in “Chapter 2-9. Verifying with the Sample Applications” on page 2-305.

Appendix B. Character Conversion

This appendix describes how DB2 handles character conversion for distributed data. The following topics are discussed in this appendix:

- “Understanding Character Conversion”
- “Specifying a System Coded Character Set Identifier” on page X-56
- “Customizing Character Conversion” on page X-62
- “When Remote Packages Should be Rebound” on page X-64.

Understanding Character Conversion

In different database management systems (DBMSs), character data can be represented by different encoding schemes. Within an encoding scheme, there are multiple coded character set identifiers (CCSIDs). EBCDIC and ASCII are two ways of encoding data. Character data transmitted from one DBMS to another might need to be converted to a different coded character set. Please be aware that ASCII support for Version 5 eliminates the need for field procedures on data in tables created within Version 5 for transactions with systems using the ASCII encoding scheme.

All character data has a CCSID. Character conversion is described in terms of CCSIDs of the source and of the target. DB2 performs most character conversion automatically, based on system CCSIDs, when data is sent to DB2 or when data is stored in DB2. You should be aware of some of the following things:

- When you install DB2 you must specify a CCSID for DB2's character data if you specify any of these values:
 - AUTO or COMMAND for the DDF STARTUP OPTION field on panel DSNTIPR
 - YES for the MIXED DATA field of panel DSNTIPF.

Which CCSID you specify depends on the national language you use.

“Specifying a System Coded Character Set Identifier” on page X-56 lists the choices available.

- Catalog table SYSIBM.SYSSTRINGS has entries for character conversions that are provided by DB2.
 - If DB2 does not provide a conversion table for a certain combination of source and target CCSIDs you will get an error message. You can add to DB2's set of conversion tables and conversion routines; “Customizing Character Conversion” on page X-62 explains how to do that.
 - If the conversion is incorrect, you might get an error message or unexpected output. To correct the problem, you need to understand the rules for assigning source and target CCSIDs in SQL operations. Chapter 3 of *SQL Reference* explains those rules.
- If you change CCSIDs or the subtypes of character columns at one DBMS, you might have to rebound packages at the other DBMS with which the first one communicates. For an explanation, see “When Remote Packages Should be Rebound” on page X-64.

Specifying a System Coded Character Set Identifier

To support character conversion, IBM's Distributed Relational Database Architecture uses CCSIDs to label the various character representation schemes.

The CCSID is a two-byte binary number that uniquely identifies one or more pairs of character sets and code pages. The coded character set defines how bit configurations are mapped for character data. A complete description of all IBM-registered CCSIDs and conversion tables exists in *Character Data Representation Architecture Registry*. For general information about character conversion, see *Character Data Representation Architecture Reference*.

The CCSID of character strings at your site is determined by the CCSID you specify on field 8 or 9 of installation panel DSNTIPF. **If this CCSID is not correct, character conversion produces incorrect results.** The correct CCSID is the number that identifies the coded character set supported by the I/O devices at your site.

If you specified MIXED DATA=NO on field 7 of installation panel DSNTIPF, you can use any valid CCSID for SBCS data. Table 127 lists a selection of common SBCS CCSIDs that might be used as source CCSIDs or target CCSIDs. DB2 does not support the storing of data into all of these CCSIDs. That is, not all of the numbers listed in Table 127 are supported as targets in conversion. To find out which combinations are supported, issue the following SQL statement:

```
General-use Programming Interface

SELECT * FROM SYSIBM.SYSSTRINGS;

End of General-use Programming Interface
```

To determine into which targets you can store data, issue the following SQL statement:

```
General-use Programming Interface

SELECT OUTCCSID FROM SYSIBM.SYSSTRINGS;

End of General-use Programming Interface
```

Table 127 (Page 1 of 3). Single-byte coded character set identifiers

Country or National Language	EBCDIC	ASCII PC	ASCII AIX	ASCII WINDOWS
# Australia (U.S. English)	37/1140*	437	819	1252/5348*
# Austria (German)	273/1141*	850/858*	819	1252/5348*
# Belgium	500/1148*	850/858*	819	1252/5348*
# Bosnia and Herzegovina (Cyrillic)	1025			1251/5347*

#

Table 127 (Page 3 of 3). Single-byte coded character set identifiers

Country or National Language	EBCDIC	ASCII PC	ASCII AIX	ASCII WINDOWS
U.S.A. (U.S. English)	37/1140*	437	819	1252/5348*

Note: * This number represents the equivalent CCSIDs using the Euro symbol.

If you specified MIXED DATA=YES on field 7 of installation panel DSNTIPF, specify a double-byte CCSID from one of the CCSIDs listed in Table 128 for EBCDIC data. If you select an ASCII encoding scheme on field 10 of installation panel DSNTIPF, then specify a CCSID for mixed data from Table 129 on page X-59. By specifying a CCSID for mixed data (an MCCSID), you also receive system CCSIDs for your SBCS and GRAPHIC data. Table 128 and Table 129 on page X-59 show the associated identifiers your site is automatically assigned when you enter a specific MCCSID. In these tables, the terms used are as follows:

- SCCSID=single-byte coded character set identifier
- MCCSID=mixed coded character set identifier
- GCCSID=graphic coded character set identifier

Table 128. EBCDIC Double-byte Coded Character Set Identifiers

National Language	MCCSID	SCCSID	GCCSID	User-defined Characters
Japanese (Extended Katakana)	930	290	300	4370
Japanese (Extended Katakana)	5026	290	4396	1880
Japanese (Extended English)	939	1027	300	4370
Japanese (Extended English)	5035	1027	4396	1880
Korean	933	833	834	1880
Simplified Chinese	935	836	837	1880
Traditional Chinese	937	28709	835	6204

Table 129. ASCII Double-byte Coded Character Set Identifiers

National Language	MCCSID	SCCSID	GCCSID	User-defined Characters
Japanese	932	897	301	1880
Japanese (Extended)	942	1041	301	1880
Japanese (Open environment)	943	897	941	1880
Japanese (HP)	5039	1041	1351	940
Korean	949	1088	951	1880
Korean (EUC)	970	367	971	188
Simplified Chinese	1381	1115	1380	1880
Simplified Chinese (EUC)	1383	367	1382	
Traditional Chinese	938	904	927	6204
Traditional Chinese	948	1043	927	6204
Traditional Chinese (IBM Big-5)	950	1114	947	

In Table 128 on page X-58 and Table 129, four CCSIDs are listed for Japanese to allow for all possible combinations of two single-byte code pages and two double-byte character sets. The difference between the single-byte code pages is in the code points for lower-case Latin letters and Katakana characters. In the code page for Japanese (Extended English, SCCSID 1027), lower-case letters have the same code points as other EBCDIC code pages.

Special considerations

Character conversion can impact performance significantly, especially if you need to process many columns or host variables.

Improving the performance of your applications: If you plan to have connections that require character conversion, you must identify character string columns with the subtype BIT DATA. Authorized users can do this by using the SQL UPDATE statement to change the FOREIGNKEY column of SYSIBM.SYSCOLUMNS to 'B' (for BIT data).

Improving the performance of applications at DBCS sites: Many of the character strings at a DBCS site are actually SBCS data, though all character string columns originating from the site are classified as MIXED DATA. To improve performance, DBCS sites that *do not* plan to have a connection between DB2 and a system using ASCII MIXED data might want to identify columns that contain only SBCS data. Authorized users can do this by using the SQL UPDATE statement to change the FOREIGNKEY column of SYSIBM.SYSCOLUMNS to 'S' (for SBCS data).

DBCS sites that *do* plan to have connections between DB2 and a system using ASCII MIXED data need to consider the lack of built-in conversions from ASCII MIXED to EBCDIC SBCS data before identifying SBCS columns. However, a site can provide its own conversion procedures for those that are not built—in. See “Customizing Character Conversion” on page X-62 for information on adding conversion procedures or rows.

Expanding Conversions: The length of a character string can increase after it undergoes conversion from one coded character set identifier to another. Such a case is known as an *expanding conversion*.

To accommodate expanding conversions, use varying length variables with a length attribute that is large enough to accommodate the expansion.

Converting to the Euro symbol

Support is provided for users wanting to migrate to CCSIDs that support the Euro symbol. This support only allows the conversion from specific CCSIDs that do not define the Euro symbol to specific CCSIDs that define the Euro symbol. See Table 130 on page X-61 or Table 131 on page X-61 for the list of CCSIDs that can be converted to the Euro symbol. In most cases, the Euro symbol replaces an existing codepoint such as the International Currency Symbol (ICS).

Altering of CCSIDs can be very disruptive to a system. Any attempt to alter CCSIDs should be performed during a maintenance window. DB2 supports only one set of CCSIDs per encoding scheme (ASCII or EBCDIC). All databases and all table spaces within an encoding scheme should be altered at the same time. DB2 does compatibility checking for SQL such as joins based on the encoding scheme. Failure to alter all databases and table spaces within an encoding scheme can result in unpredictable results.

Before attempting to alter the CCSID of a system, the following information must be obtained.

- # 1. Current CCSID
- # 2. Planned CCSID
- # 3. List of databases created with current CCSID
- # 4. List of table spaces created with current CCSID
- # 5. List of views defined on tables defined in the table spaces that will have the CCSID altered
- # 6. Definitions of those views
- # 7. Authorization information on the views

The method for changing the CCSIDs requires that the data be unloaded and reloaded into DB2. During the load phase, the data should be converted using the CCSID and NOSUBS keywords (added in APAR PQ22904) from the old CCSID to the new CCSID. Any records not converting cleanly will be placed in a discard data set.

- # 1. Modify the CCSID data in DSNHDECP by running the installation CLIST and specifying UPDATE on installation panel DSNTIPA1. From panel DSNTIPB, select Application Programming Defaults Panel 1. For details on the update process, see "The Update Process" on page 2-228.
- # 2. Run job DSNTIJUZ created in the above step.
- # 3. Stop and restart DB2 to use the new parameters.
- # 4. Alter databases. This will only affect the default for new table spaces created in the altered database.
- # 5. Drop any views on tables that exist in any table space that you want to alter.

- # 6. Unload data from the tables that you want to alter.
- # 7. Alter the CCSIDs on the table spaces. This will invalidate any plans or packages that reference these table spaces.
- # 8. Reload data from tables.
- # 9. Recreate views.
- # 10. Recreate authorizations on the views.
- # 11. Rebind the invalidated plans and packages either with autobind or manually.

The list of CCSIDs that can be modified are listed according to the encoding scheme:

Table 130. EBCDIC CCSID values that convert to Euro CCSIDs

CCSIDs without Euro symbol	CCSIDs with Euro symbol
37	1140
273	1141
277	1142
278	1143
280	1144
284	1145
285	1146
297	1147
500	1148
871	1149

Table 131. ASCII CCSID values that convert to Euro CCSIDs

CCSIDs without Euro symbol	CCSIDs with Euro symbol
850	858
1250	5346
1251	5347
1252	5348
1253	5349
1254	5350
1255	5351
1256	5352
1257	5353
874	4970

You can not convert other CCSIDs to the Euro supported CCSIDs. All databases and all table spaces within an encoding scheme should be altered at the same time.

Customizing Character Conversion

The SYSIBM.SYSSTRINGS catalog table contains entries that support most single-byte and double-byte character conversions. If a conversion is not included in SYSIBM.SYSSTRINGS, it is not automatically performed by DB2. To determine which conversions DB2 supports, execute this query:

```
SELECT * FROM SYSIBM.SYSSTRINGS;
```

If you need to add a conversion that is not provided by DB2 or override a built-in conversion, you must modify the catalog table SYSIBM.SYSSTRINGS. Any user with SYSADM, SYSCTRL, or DBADM authority for the catalog database has SELECT, INSERT, UPDATE, and DELETE privileges on SYSSTRINGS, subject to validation by an IBM-provided validation procedure. See “Rules for SYSSTRINGS Entries” on page X-64 for more information on this validation procedure.

How an Entry in SYSIBM.SYSSTRINGS Works

The catalog table SYSIBM.SYSSTRINGS contains the following columns:

INCCSID	The source CCSID of a character conversion.
OUTCCSID	The target CCSID of a character conversion.
TRANSTYPE	The type of conversion: <ul style="list-style-type: none">SS SBCS data to SBCS dataSM SBCS data to EBCDIC MIXED dataMS EBCDIC MIXED data to SBCS (EBCDIC and ASCII) dataPS ASCII MIXED data to SBCS (EBCDIC and ASCII) dataGG GRAPHIC data to GRAPHIC dataPM ASCII MIXED data to EBCDIC MIXED dataMM EBCDIC MIXED data to EBCDIC MIXED data.MP EBCDIC MIXED to ASCII MIXED data.PP ASCII MIXED to ASCII MIXED data.SP SBCS (ASCII and EBCDIC) to ASCII MIXED data.
ERRORBYTE	Specifies the byte used in the conversion table (TRANSTAB) as an error indicator. For example, if ERRORBYTE is X'3E', that byte is used in the conversion table to indicate that no conversion is defined for code points that map to X'3E'. Null indicates the absence of an error indicator.
SUBBYTE	Specifies the byte used in the conversion table (TRANSTAB) as a substitution character. For example, if SUBBYTE is X'3F', that byte is used in the conversion table as a substitute for code points that map to X'3F'. A warning occurs when a code point maps to the value of SUBBYTE. Null indicates the absence of a substitution character.
TRANSPROC	The name of a module or a blank string. If IBMREQD is N, a non-blank value of TRANSPROC is the name of a user-provided conversion procedure. If IBMREQD is Y, a non-blank value of TRANSPROC is the name of a DB2 module that contains DBCS conversion tables.
IBMREQD	Y indicates that the row is provided by IBM. N indicates that the row has been inserted by the user.
TRANSTAB	A 256-byte conversion table or an empty string.

Each row of SYSSTRINGS contains information about the conversion of character strings from the coded character set identified by INCCSID to the coded character set identified by OUTCCSID. The conversion function is automatically invoked when a conversion from the coded character set identified by the INCCSID column to the coded character set identified by the OUTCCSID column is required.

For example, the row of SYSSTRINGS in which the value of INCCSID is 500 and the value of OUTCCSID is 37 describes the conversion from CCSID 500 to CCSID 37. The row in which the value of INCCSID is 37 and the value of OUTCCSID is 500 describes the conversion from CCSID 37 to CCSID 500.

DB2 enforces a distinction between IBM-supplied rows and user-provided rows with the following constraints:

- Rows with IBMREQD=Y cannot be updated or deleted.
- Rows with IBMREQD=N can be inserted, updated, and deleted.
- The same pair of CCSIDs can be in two rows, provided one is in an IBM-supplied row and the other is in a user-provided row. In this case, the user-provided row is used for the character conversion.

The following types of rows are possible in SYSSTRINGS:

- TRANSPROC is blank and TRANSTAB is an empty string. This indicates that no conversion is performed.
- IBMREQD=N and TRANSPROC is not blank. This indicates that the conversion is performed by a conversion procedure that you provide; the conversion procedure module name is identified in the TRANSPROC column.
- Neither of the above. This indicates that the conversion is performed by a DB2 module. For example, if TRANSPROC is blank and TRANSTAB is not empty, the conversion function uses the conversion table identified in TRANSTAB.

Adding a Character Conversion Procedure to DB2

For SBCS conversions that do not involve expansion or contraction, you can provide additional conversions by adding a row to SYSSTRINGS that contains a new conversion table in the TRANSTAB column. The 256-byte conversion table contained in TRANSTAB provides simple character conversion by using the numerical value of a character as a displacement into the conversion table to get the replacement character. If the row you have inserted into SYSSTRINGS describes an SBCS conversion and TRANSPROC is blank, the DB2 conversion module will perform the conversion using the conversion table contained in TRANSTAB.

You can use sample job DSNTJ1T to add rows to SYSSTRINGS.

When you add a user-provided row to SYSSTRINGS to override an existing row for a certain CCSID pair, the change does not take effect until DB2 is restarted. Likewise, updating or deleting a user-provided row does not take effect until DB2 is restarted. However, if you add a new row for a CCSID pair that did not previously have a row in SYSSTRINGS, the new row is used without restarting DB2 if there is a request for that CCSID pair.

For DBCS conversions, the TRANSPROC column of the inserted row must identify a user-provided conversion procedure. A user-provided conversion procedure can also be identified for SBCS conversions, but it is not necessary.

The rules for writing conversion procedures are similar to the rules for writing other DB2 installation-wide exit routines. For more information on writing exit routines for character conversion see Appendixes (Volume 2) of *Administration Guide* .

Rules for SYSSTRINGS Entries

- An INSERT, UPDATE, DELETE, or LOAD is allowed only if IBMREQD = N.
- The values in the INCCSID and OUTCCSID columns must be in the range of 1-65533.
- For any given row, the INCCSID and OUTCCSID columns cannot contain the same value.
- The value in the TRANSTYPE column must be SS, SM, MS, PS, MM, PM, GG, MP, PP, or SP.
- For any given row, the ERRORBYTE and SUBBYTE columns cannot contain the same nonnull value.
- The TRANSPROC column must either be blank or contain a string that conforms to the rules for MVS program names.
- The length specified in the TRANSTAB column must be either 0 or 256.

When Remote Packages Should be Rebound

Certain conversion-related changes at the local or a remote DBMS may force the rebinding of a remotely bound package. These include the following:

- The system CCSID at the remote DBMS was changed. In this case, the package should always be rebound.
- The system CCSID at the local DBMS was changed. This could happen, for example, if the wrong system CCSID was specified during installation. If so, string constants in static SQL statements might have been converted incorrectly during the binding of the package. Rebinding will correct the conversion. Other problems could also arise as a result of the change. Indeed, it might be prudent to always rebind.
- The subtype of a character column is changed at the remote DBMS. The pertinent changes are from BIT to either SBCS or MIXED, and from SBCS or MIXED to BIT. The change would have been made by modifying the FOREIGNKEY column of the SYSIBM.SYSCOLUMNS table in the remote system catalog. Or, the change could have occurred if the table was dropped and re-created with a different subtype (BIT to either SBCS or MIXED, or vice versa) after the application was bound. A statement that refers to a column with a modified subtype can begin to fail with an SQLCODE of -333. If this occurs, the package containing the statement should be rebound.

Glossary and Bibliography

Glossary

The following terms and abbreviations are defined as they are used in the DB2 library. If you do not find the term you are looking for, refer to the index or to *Dictionary of Computing*.

A

abend. Abnormal end of task.

abend reason code. A 4-byte hexadecimal code that uniquely identifies a problem with DB2. A complete list of DB2 abend reason codes and their explanations is contained in *Messages and Codes*.

abnormal end of task (abend). Termination of a task, a job, or a subsystem because of an error condition that cannot be resolved during execution by recovery facilities.

access method services. A utility program that defines and manages VSAM data sets (or files).

access path. The path used to get to data specified in SQL statements. An access path can involve an index or a sequential search.

active log. The portion of the DB2 log to which log records are written as they are generated. The active log always contains the most recent log records, whereas the archive log holds those records that are older and no longer will fit on the active log.

address space. A range of virtual storage pages identified by a number (ASID) and a collection of segment and page tables which map the virtual pages to real pages of the computer's memory.

address space connection. The result of connecting an allied address space to DB2. Each address space containing a task connected to DB2 has exactly one address space connection, even though more than one task control block (TCB) can be present. See *allied address space* and *task control block*.

alias. An alternate name that can be used in SQL statements to refer to a table or view in the same or a remote DB2 subsystem.

allied address space. An area of storage external to DB2 that is connected to DB2 and is therefore capable of requesting DB2 services.

allied thread. A thread originating at the local DB2 subsystem that may access data at a remote DB2 subsystem.

already verified. An LU 6.2 security option which allows DB2 to provide the user's verified authorization ID when allocating a conversation. The user is not validated by the partner DB2.

ambiguous cursor. A database cursor that is not defined with either the clauses FOR FETCH ONLY or FOR UPDATE OF, is not defined on a read-only result table, is not the target of a WHERE CURRENT clause on an SQL UPDATE or DELETE statement, and is in a plan or package that contains SQL statements PREPARE or EXECUTE IMMEDIATE.

APAR. Authorized program analysis report.

APAR fix corrective service. A temporary correction of a DB2 defect. The correction is temporary because it is usually replaced at a later date by a more permanent correction such as a program temporary fix (PTF).

APF. Authorized program facility.

API. Application programming interface.

APPL. A VTAM network definition statement used to define DB2 to VTAM as an application program using SNA LU 6.2 protocols.

application. A program or set of programs that perform a task; for example, a payroll application.

application plan. The control structure produced during the bind process and used by DB2 to process SQL statements encountered during statement execution.

application process. The unit to which resources and locks are allocated. An application process involves the execution of one or more programs.

application program interface (API). A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or licensed program.

application requester (AR). See *requester*.

application server. See *server*.

AR. application requester. See *requester*.

archive log. The portion of the DB2 log that contains log records that have been copied from the active log.

AS • character string

AS. Application server. See *server*.

ASCII. An encoding scheme used to represent strings in many environments, typically on PCs and workstations. Contrast with *EBCDIC*.

attachment facility. An interface between DB2 and TSO, IMS, CICS, or batch address spaces. An attachment facility allows application programs to access DB2.

attribute. A characteristic of an entity. For example, in database design, the phone number of an employee is one of that employee's attributes.

authorization ID. A string that can be verified for connection to DB2 and to which a set of privileges are allowed. It can represent an individual, an organizational group, or a function, but DB2 does not determine this representation.

authorized program analysis report (APAR). A report of a problem caused by a suspected defect in a current release of a program.

authorized program facility (APF). A facility that permits the identification of programs that are authorized to use restricted functions.

B

backward log recovery. The fourth and final phase of restart processing during which DB2 scans the log in a backward direction to apply UNDO log records for all aborted changes.

base table. A table created by the SQL CREATE TABLE statement that is used to hold persistent data. Contrast with *result table* and *temporary table*.

basic sequential access method (BSAM). An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device.

bind. The process by which the output from the DB2 precompiler is converted to a usable control structure called a package or an application plan. During the process, access paths to the data are selected and some authorization checking is performed.

automatic bind. (More correctly *automatic rebind*). A process by which SQL statements are bound automatically (without a user issuing a BIND command) when an application process begins execution and the bound application plan or package it requires is not valid.

dynamic bind. A process by which SQL statements are bound as they are entered.

incremental bind. A process by which SQL statements are bound during the execution of an application process, because they could not be bound during the bind process, and VALIDATE(RUN) was specified.

static bind. A process by which SQL statements are bound after they have been precompiled. All static SQL statements are prepared for execution at the same time. Contrast with *dynamic bind*.

BMP. Batch Message Processing (IMS).

bootstrap data set (BSDS). A VSAM data set that contains name and status information for DB2, as well as RBA range specifications, for all active and archive log data sets. It also contains passwords for the DB2 directory and catalog, and lists of conditional restart and checkpoint records.

BSAM. Basic sequential access method.

BSDS. Bootstrap data set.

buffer pool. Main storage reserved to satisfy the buffering requirements for one or more table spaces or indexes.

built-in function. Scalar function or column function.

C

CAF. Call attachment facility.

call attachment facility (CAF). A DB2 attachment facility for application programs running in TSO or MVS batch. The CAF is an alternative to the DSN command processor and allows greater control over the execution environment.

cascade delete. The enforcement of referential constraints by DB2 when it deletes all descendent rows of a deleted parent row.

catalog. In DB2, a collection of tables that contains descriptions of objects such as tables, views, and indexes.

catalog table. Any table in the DB2 catalog.

CCSID. Coded character set identifier.

CDB. See *communications database*.

character set. A defined set of characters.

character string. A sequence of bytes representing bit data, single-byte characters, or a mixture of single and double-byte characters.

check clause. An extension to the SQL CREATE TABLE and SQL ALTER TABLE statements that specifies a table check constraint.

check constraint. See *table check constraint*.

check integrity. The condition that exists when each row in a table conforms to the table check constraints defined on that table. Maintaining check integrity requires enforcing table check constraints on operations that add or change data.

check pending. A state of a table space or partition that prevents its use by some utilities and some SQL statements, because it can contain rows that violate referential constraints, table check constraints, or both.

checkpoint. A point at which DB2 records internal status information on the DB2 log that would be used in the recovery process if DB2 should abend.

CI. Control interval.

CICS. Represents (in this publication) CICS/MVS and CICS/ESA.

CICS/MVS: Customer Information Control System/Multiple Virtual Storage.

CICS/ESA: Customer Information Control System/Enterprise Systems Architecture.

CICS attachment facility. A DB2 subcomponent that uses the MVS Subsystem Interface (SSI) and cross storage linkage to process requests from CICS to DB2 and to coordinate resource commitment.

CIDF. Control interval definition field.

claim. To register to DB2 that an object is being accessed. This registration is also called a claim. A claim is used to ensure that an object cannot be drained until a commit is reached. Contrast with *drain*.

claim class. A specific type of object access which can be one of the following:

- cursor stability (CS)
- repeatable read (RR)
- write

claim count. A count of the number of agents that are accessing an object.

class of service. A VTAM term for a list of routes through a network, arranged in an order of preference for their use.

clause. In SQL, a distinct part of a statement, such as a SELECT clause or a WHERE clause.

client. See *requester*.

CLIST. Command list. A language for performing TSO tasks.

CLPA. Create link pack area.

clustering index. An index that determines how rows are physically ordered in a table space.

coded character set. A set of unambiguous rules that establish a character set and the one-to-one relationships between the characters of the set and their coded representations.

coded character set identifier (CCSID). A 16-bit number that uniquely identifies a coded representation of graphic characters. It designates an encoding scheme identifier and one or more pairs consisting of a character set identifier and an associated code page identifier.

column. The vertical component of a table. A column has a name and a particular data type (for example, character, decimal, or integer).

column function. An SQL operation that derives its result from a collection of values across one or more rows. Contrast with *scalar function*.

"come from" checking. An LU 6.2 security option which defines a list of authorization IDs that are allowed to connect to DB2 from a partner LU.

command. A DB2 operator command or a DSN subcommand. Distinct from an SQL statement.

command recognition character (CRC). A character that permits an MVS console operator or an IMS subsystem user to route DB2 commands to specific DB2 subsystems.

commit. The operation that ends a unit of work by releasing locks so that the database changes made by that unit of work can be perceived by other processes.

commit point. A point in time when data is considered consistent.

committed phase. The second phase of the multi-site update process that requests all participants to commit the effects of the logical unit of work.

common service area (CSA). In MVS, a part of the common area that contains data areas addressable by all address spaces.

communications database (CDB). A set of tables in the DB2 catalog that are used to establish conversations with remote database management systems.

comparison operator • database descriptor (DBD)

comparison operator. A token (such as =, >, <) used to specify a relationship between two values.

compression dictionary. The dictionary that controls the process of compression and decompression. This dictionary is created from the data in the table space or table space partition.

concurrency. The shared use of resources by more than one application process at the same time.

conditional restart. A DB2 restart that is directed by a user-defined conditional restart control record (CRCR).

connection ID. An identifier supplied by the attachment facility that is associated with a specific address space connection.

consistency token. A timestamp used to generate the version identifier for an application. See also *version*.

constraint. A rule that limits the values that can be inserted, deleted, or updated in a table. See *referential constraint*, *uniqueness constraint*, and *table check constraint*.

control interval (CI). A fixed-length area or direct access storage in which VSAM stores records and creates distributed free space. Also, in a key-sequenced data set or file, the set of records pointed to by an entry in the sequence-set index record. The control interval is the unit of information that VSAM transmits to or from direct access storage. A control interval always includes an integral number of physical records.

control interval definition field (CIDF). In VSAM, a field located in the four bytes at the end of each control interval; it describes the free space, if any, in the control interval.

conversation. (1) A VTAM term for a dialog between two application processes, on different DB2 subsystems, that is specified by a particular *session name*, *mode name*, and *LU name*. (2) An LU 6.2 security option which allows DB2 to require the user's authorization ID and password when allocating a conversation to a partner DB2. The user is validated by the partner DB2.

coordinator. The system component that coordinates the commit or rollback of a unit of work that includes work done on one or more other systems.

correlated subquery. A subquery (part of a WHERE or HAVING clause) applied to a row or group of rows of a table or view named in an outer sub-SELECT statement.

correlation ID. An identifier associated with a specific thread. In TSO, it is either an authorization ID or the job name.

correlation name. An identifier that designates a table, a view, or individual rows of a table or view within a single SQL statement. It can be defined in any FROM clause or in the first clause of an UPDATE or DELETE statement.

CRC. Command recognition character.

CRCR. Conditional restart control record.

cross-memory linkage. A method for invoking a program in a different address space. The invocation is synchronous with respect to the caller.

CSA. Common service area.

current status rebuild. The second phase of restart processing during which the status of the subsystem is reconstructed from information on the log.

cursor. A named control structure used by an application program to point to a row of interest within some set of rows, and to retrieve rows from the set, possibly making updates or deletions.

cursor stability (CS). The isolation level that provides maximum concurrency without the ability to read uncommitted data. With cursor stability, a unit of work holds locks only on its uncommitted changes and on the current row of each of its cursors.

cursor table (CT). The cursor table is the copy of the skeleton cursor table used by an executing application process.

cycle. A set of tables that can be ordered so that each table is a descendent of the one before it, and the first is a descendent of the last. A self-referencing table is a cycle with a single member.

D

DASD. Direct access storage device.

database. A collection of tables, or a collection of table spaces and index spaces.

database access thread. A thread accessing data at the local subsystem on behalf of a remote subsystem.

database administrator (DBA). An individual responsible for the design, development, operation, safeguarding, maintenance, and use of a database.

database descriptor (DBD). An internal representation of DB2 database definition which reflects the data definition found in the DB2 catalog. The objects defined in a database descriptor are table

spaces, tables, indexes, index spaces, and relationships.

database management system (DBMS). A software system that controls the creation, organization, and modification of a database and access to the data stored within it.

database request module (DBRM). A data set member created by the DB2 precompiler that contains information about SQL statements. DBRMs are used in the bind process.

DATABASE 2 Interactive (DB2I). The DB2 facility that provides for the execution of SQL statements, DB2 (operator) commands, programmer commands, and utility invocation.

data definition name (DD name). The name of a data definition (DD) statement that corresponds to a data control block containing the same name.

Data Language/I (DL/I). The IMS data manipulation language; a common high-level interface between a user application and IMS.

data type. An attribute of columns, literals, host variables, special registers, and the results of functions and expressions.

date. A three-part value that designates a day, month, and year.

date duration. A decimal integer that represents a number of years, months, and days.

datetime value. A value of the data type DATE, TIME, or TIMESTAMP.

DBA. Database administrator.

DBCS. Double-byte character set.

DBD. Database descriptor.

DBID. Database identifier.

DBMS. Database management system.

DBRM. Database request module.

DB2 catalog. Tables maintained by DB2 that contain descriptions of DB2 objects such as tables, views, and indexes.

DB2 command. An instruction to the DB2 subsystem allowing a user to start or stop DB2, to display information on current users, to start or stop databases, to display information on the status of databases, and so on.

DB2I. DATABASE 2 Interactive.

DB2I Kanji Feature. The tape that contains the panels and jobs that allow a site to display DB2I panels in Kanji.

DB2 PM. DATABASE 2 Performance Monitor.

DB2 private protocol access. A method of accessing distributed data by which you can direct a query to another DB2 system by using an alias or a three-part name to identify the DB2 subsystems at which the statements are executed. Contrast with *DRDA access*.

DB2 private protocol connection. A DB2 private connection of the application process. See also *private connection*.

DCLGEN. Declarations generator.

DDF. Distributed data facility.

DD name. Data definition name.

deadlock. Unresolvable contention for the use of a resource such as a table or an index.

declarations generator (DCLGEN). A subcomponent of DB2 that generates SQL table declarations and COBOL, C, or PL/I data structure declarations that conform to the table. The declarations are generated from DB2 system catalog information. DCLGEN is also a DSN subcommand.

default value. A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

delete rule. The rule that tells DB2 what to do to a dependent row when a parent row is deleted. For each relationship, the rule might be CASCADE, RESTRICT, SET NULL, or NO ACTION.

dependent. An object (row, table, or table space) is a dependent if it has at least one parent. The object is also said to be a dependent (row, table, or table space) of its parent. See *parent row*, *parent table*, *parent table space*.

dependent row. A row that contains a foreign key that matches the value of a primary key in the parent row.

dependent table. A table that is dependent in at least one referential constraint.

descendent. An object is a descendent of another object if it is a dependent of the object, or if it is the dependent of a descendent of that object.

descendent row. A row that is dependent on another row or a row that is a dependent of a descendent row.

descendent table • expression

descendent table. A table that is a dependent of another table or a dependent of a descendent table.

DFHSM. Data Facility Hierarchical Storage Manager.

DFP. Data Facility Product (MVS).

direct access storage device (DASD). A device in which access time is independent of the location of the data.

directory. The system database that contains internal objects such as database descriptors and skeleton cursor tables.

distributed data facility (DDF). A set of DB2 components through which DB2 communicates with another RDBMS.

distributed relational database architecture (DRDA). A connection protocol for distributed relational database processing that is used by IBM's relational database products. DRDA includes protocols for communication between an application and a remote relational database management system, and for communication between relational database management systems.

DL/I. Data Language/I. The IMS data manipulation language; a common high-level interface between a user application and IMS.

double-byte character set (DBCS). A set of characters used by national languages such as Japanese and Chinese that have more symbols than can be represented by a single byte. Each character is two bytes in length, and therefore requires special hardware to be displayed or printed.

drain. To acquire a locked resource by quiescing access to that object.

drain lock. A lock on a claim class which prevents a claim from occurring.

DRDA. Distributed relational database architecture.

DRDA access. A method of accessing distributed data by which you can explicitly connect to another location, using an SQL statement, to execute packages that have been previously bound at that location. The SQL CONNECT statement is used to identify application servers, and SQL statements are executed using packages that were previously bound at those servers. Contrast with *DB2 private protocol access*.

DSN. (1) The default DB2 subsystem name. (2) The name of the TSO command processor of DB2. (3) The first three characters of DB2 module and macro names.

duration. A number that represents an interval of time. See *date duration*, *labeled duration*, and *time duration*.

dynamic SQL. SQL statements that are prepared and executed within an application program while the program is executing. In dynamic SQL, the SQL source is contained in host language variables rather than being coded into the application program. The SQL statement can change several times during the application program's execution.

E

EBCDIC. Extended binary coded decimal interchange code. An encoding scheme used to represent character data in the MVS, VM, VSE, and OS/400 environments. Contrast with *ASCII*.

EDM pool. A pool of main storage used for database descriptors and application plans.

EID. Event identifier.

embedded SQL. SQL statements coded within an application program. See *static SQL*.

EOM. End of memory.

EOT. End of task.

error page range. Range of pages considered to be physically damaged. DB2 will not allow a user to access any pages that fall within this range.

equi-join. A join operation in which the join-condition has the form *expression = expression*.

ESDS. Entry sequenced data set.

ESMT. External subsystem module table (IMS).

EUR. IBM European Standards.

exception table. A table that holds rows that violate referential constraints or table check constraints found by the CHECK DATA utility.

exclusive lock. A lock that prevents concurrently executing application processes from reading or changing data. Contrast with *shared lock*.

exit routine. A user-written (or IBM-provided default) program that receives control from DB2 to perform specific functions. Exit routines run as extensions of DB2.

expression. An operand or a collection of operators and operands that yields a single value.

F

fallback. The process of returning to a previous release of DB2 after attempting or completing migration to a current release.

field procedure. A user-written exit routine designed to receive a single value and transform (encode or decode) it in any way the user can specify.

fixed-length string. A character or graphic string whose length is specified and cannot be changed. Contrast with *varying-length string*.

foreign key. A key that is specified in the definition of a referential constraint. Because of the foreign key, the table is a dependent table. The key must have the same number of columns, with the same descriptions, as the primary key of the parent table.

forward log recovery. The third phase of restart processing during which DB2 processes the log in a forward direction to apply all REDO log records.

free space. The total unused space in a page, that is, the space not used to store records or control information.

full outer join. The result of a join operation that includes the matched rows of both tables being joined and preserves the unmatched rows of both tables. See also *join*.

function. A scalar function or column function. Same as *built-in function*.

G

GB. Gigabyte (1,073,741,824 bytes).

generalized trace facility (GTF). An MVS service program that records significant system events such as I/O interrupts, SVC interrupts, program interrupts, or external interrupts.

getpage. An operation in which DB2 accesses a data page.

GIMSMP. The load module name for the System Modification Program/Extended, a basic tool for installing, changing, and controlling changes to programming systems.

graphic string. A sequence of DBCS characters.

gross lock. The *shared*, *update*, or *exclusive* mode locks on a table, partition, or table space.

group buffer pool. A coupling facility cache structure used by a data sharing group to cache data and to ensure that the data is consistent for all members.

GTF. Generalized trace facility.

H

help panel. A screen of information presenting tutorial text to assist a user at the terminal.

home address space. The area of storage that MVS currently recognizes as “dispatched.”

host language. A programming language in which you can embed SQL statements.

host program. An application program written in a host language that contains embedded SQL statements.

host structure. In an application program, a structure referenced by embedded SQL statements.

host variable. In an application program, an application variable referenced by embedded SQL statements.

HSM. Hierarchical storage manager.

I

ICF. Integrated catalog facility.

IDCAMS. An IBM program used to process access method services (AMS) commands. It can be invoked as a job or jobstep, from a TSO terminal, or from within a user's application program.

IDCAMS LISTCAT. A facility for obtaining information contained in the access method services catalog.

identify. A request that an attachment service program in an address space separate from DB2 issues via the MVS subsystem interface to inform DB2 of its existence and initiate the process of becoming connected to DB2.

IFI. Instrumentation facility interface.

IFI call. An invocation of the instrumentation facility interface (IFI) by means of one of its defined functions.

IFP. IMS Fast Path.

image copy. An exact reproduction of all or part of a table space. DB2 provides utility programs to make full image copies (to copy the entire table space) or incremental image copies (to copy only those pages that have been modified since the last image copy).

IMS. Information Management System.

IMS attachment facility • KSDS

IMS attachment facility. A DB2 subcomponent that uses MVS Subsystem Interface (SSI) protocols and cross-memory linkage to process requests from IMS to DB2 and to coordinate resource commitment.

IMS DB. Information Management System Database.

IMS TM. Information Management System Transaction Manager.

in-abort. A status of a unit of recovery. If DB2 fails after a unit of recovery begins to be rolled back, but before the process is completed, DB2 will continue to back out the changes during restart.

in-commit. A status of a unit of recovery. If DB2 fails after beginning its phase 2 commit processing, it “knows,” when restarted, that changes made to data are consistent. Such units of recovery are termed *in-commit*.

independent. An object (row, table, or table space) is independent if it is neither a parent nor a dependent of another object.

index. A set of pointers that are logically ordered by the values of a key. Indexes can provide faster access to data and can enforce uniqueness on the rows in a table.

index key. The set of columns in a table used to determine the order of index entries.

index space. A page set used to store the entries of one index.

indicator variable. A variable used to represent the null value in an application program. If the value for the selected column is null, a negative value is placed in the indicator variable.

indoubt. A status of a unit of recovery. If DB2 fails after it has finished its phase 1 commit processing and before it has started phase 2, only the commit coordinator knows if this unit of recovery is to be committed or rolled back. At emergency restart, if DB2 does not have the information needed to make this decision, its unit of recovery is *indoubt* until DB2 obtains this information from the coordinator.

indoubt resolution. The process of resolving the status of an indoubt logical unit of work to either the committed or the rollback state.

inflight. A status of a unit of recovery. If DB2 fails before its unit of recovery completes phase 1 of the commit process, it merely backs out the updates of its

unit of recovery when it is restarted. These units of recovery are termed *inflight*.

inner join. The result of a join operation that includes only the matched rows of both tables being joined. See also *join*.

install. The process of preparing a DB2 subsystem to operate as an MVS subsystem.

installation verification scenario. A sequence of operations that exercises the main DB2 functions and tests whether DB2 was correctly installed.

Interactive System Productivity Facility (ISPF). An IBM licensed program that provides interactive dialog services.

internal resource lock manager (IRLM). An MVS subsystem used by DB2 to control communication and database locking.

IRLM. internal resource lock manager.

ISO. International Standards Organization.

isolation level. The degree to which a unit of work is isolated from the updating operations of other units of work. See also *cursor stability*, *repeatable read*, *uncommitted read*, and *read stability*.

ISPF. Interactive System Productivity Facility.

ISPF/PDF. Interactive System Productivity Facility/Program Development Facility.

J

JCL. Job control language.

JES. MVS Job Entry Subsystem.

JIS. Japanese Industrial Standard.

join. A relational operation that allows retrieval of data from two or more tables based on matching column values. See also *full outer join*, *inner join*, *left outer join*, *outer join*, *right outer join*, *equi-join*.

K

KB. Kilobyte (1024 bytes).

key. A column or an ordered collection of columns identified in the description of a table, index, or referential constraint.

KSDS. Key sequenced data set.

L

labeled duration. A number that represents a duration of years, months, days, hours, minutes, seconds, or microseconds.

latch. A DB2 internal mechanism for controlling concurrent events or the use of system resources.

LCID. Log control interval definition.

LDS. Linear data set.

leaf page. A page that contains pairs of keys and RIDs and that points to actual data. Contrast with *nonleaf page*.

left outer join. The result of a join operation that includes the matched rows of both tables being joined, and preserves the unmatched rows of the first table. See also *join*.

linear data set (LDS). A VSAM data set that contains data but no control information. A linear data set can be accessed as a byte-addressable string in virtual storage.

link-edit. To create a loadable computer program using a linkage editor.

L-lock. See *logical lock*.

load module. A program unit that is suitable for loading into main storage for execution. The output of a linkage editor.

local subsystem. The unique RDBMS to which the user or application program is directly connected (in the case of DB2, by one of the DB2 attachment facilities).

lock. A means of controlling concurrent events or access to data. DB2 locking is performed by the IRLM.

lock duration. The interval over which a DB2 lock is held.

lock escalation. The promotion of a lock from a row or page lock to a table space lock because the number of page locks concurrently held on a given resource exceeds a preset limit.

locking. The process by which the integrity of data is ensured. Locking prevents concurrent users from accessing inconsistent data.

lock mode. A representation for the type of access concurrently running programs can have to a resource held by a DB2 lock.

lock object. The resource that is controlled by a DB2 lock.

lock promotion. The process of changing the size or mode of a DB2 lock to a higher level.

lock size. The amount of data controlled by a DB2 lock on table data; the value can be a row, a page, a table, or a table space.

log. A collection of records that describe the events that occur during DB2 execution and their sequence. The information thus recorded is used for recovery in the event of a failure during DB2 execution.

logical lock. The lock type used by transactions to control intra- and inter-DB2 data concurrency between transactions.

logical recovery pending (LRECP). The state in which the data and the index keys that reference the data are inconsistent.

logical unit. An access point through which an application program accesses the SNA network in order to communicate with another application program.

logical unit of work (LUW). In IMS, the processing that program performs between synchronization points.

logical unit of work identifier (LUWID). A name that uniquely identifies a thread within a network. This name consists of a fully-qualified LU network name, an LUW instance number, and an LUW sequence number.

log initialization. The first phase of restart processing during which DB2 attempts to locate the current end of the log.

log record sequence number (LRSN). A number DB2 generates and associates with each log record. DB2 also uses the LRSN for page versioning. The LRSNs generated by a given DB2 data sharing group form a strictly increasing sequence for each DB2 log and a strictly increasing sequence for each page across the DB2 group.

log truncation. A process by which an explicit starting RBA is established. This RBA is the point at which the next byte of log data will be written.

long string. A string whose actual length, or a varying-length string whose maximum length, is greater than 255 bytes or 127 double-byte characters.

LRECP. Logical recovery pending.

LRH. Log record header.

LRSN. See *log record sequence number*.

LUW. Logical unit of work.

LUID • page set recovery pending (PSRCP)

LUID. Logical unit of work identifier.

M

MB. Megabyte (1,048,576 bytes).

migration. The process of converting a DB2 subsystem with a previous release of DB2 to an updated or current release. In this process, you can acquire the functions of the updated or current release without losing the data you created on the previous release.

mixed data string. A character string that can contain both single-byte and double-byte characters.

MLPA. Modified link pack area.

MODEENT. A VTAM macro instruction which associates a logon mode name with a set of parameters representing session protocols. A set of MODEENT macro instructions defines a logon mode table.

mode name. A VTAM name for the collection of physical and logical characteristics and attributes of a *session*.

MPP. Message processing program (IMS).

MSS. Mass Storage Subsystem

MTO. Master terminal operator.

multi-site update. Distributed relational database processing in which data is updated in more than one location within a single unit of work.

must-complete. A state during DB2 processing in which the entire operation must be completed to maintain data integrity.

MVS. Multiple Virtual Storage.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture.

MVS/XA. Multiple Virtual Storage/Extended Architecture.

N

NID (network identifier). The network ID assigned by IMS or CICS, or if the connection type is RRSAF, the OS/390 RRS Unit of Recovery ID (URID).

nonleaf page. A page that contains keys and page numbers of other pages in the index (either leaf or nonleaf pages). Nonleaf pages never point to actual data.

NRE. Network recovery element.

NUL. In C, a single character that denotes the end of the string.

null. A special value that indicates the absence of information.

NUL-terminated host variable. A varying-length host variable in which the end of the data is indicated by the presence of a NUL terminator.

NUL terminator. In C, the value that indicates the end of a string. For character strings, the NUL terminator is 'X'00'.

O

OASN (origin application schedule number). In IMS, a 4-byte number assigned sequentially to each IMS schedule since the last cold start of IMS and used as an identifier for a unit of work. In an 8-byte format, the first four bytes contain the schedule number and the last four contain the number of IMS sync points (*commit points*) during the current schedule. The OASN is part of the NID for an IMS connection.

OBID. Data object identifier.

outer join. The result of a join operation that includes the matched rows of both tables being joined and preserves some or all of the unmatched rows of the tables being joined. See also *join*.

P

package. Also *application package*. An object containing a set of SQL statements that have been bound statically and that are available for processing.

package list. An ordered list of package names that may be used to extend an application plan.

package name. The name given an object created by a BIND PACKAGE or REBIND PACKAGE command. The object is a bound version of a database request module (DBRM). The name consists of a location name, a collection ID, a package ID, and a version ID.

page. A unit of storage within a table space (4KB or 32KB) or index space (4KB). In a table space, a page contains one or more rows of a table.

page set. A table space or index space consisting of pages that are either 4KB or 32KB in size. Each page set is made from a collection of VSAM data sets.

page set recovery pending (PSRCP). A restrictive state of an index space in which the page set is in a

recovery pending state. In this case, the entire page set must be recovered. Recovery of a logical part is prohibited.

parallel I/O processing. A form of I/O processing in which DB2 initiates multiple concurrent requests for a single user query and performs I/O processing concurrently (in *parallel*), on multiple data partitions.

parent row. A row whose primary key value is the foreign key value of a dependent row.

parent table. A table whose primary key is referenced by the foreign key of a dependent table.

parent table space. A table space that contains a parent table. A table space containing a dependent of that table is a dependent table space.

participant. An entity other than the commit coordinator that takes part in the commit process. Synonymous with *agent* in SNA.

partition. A portion of a page set. Each partition corresponds to a single, independently extendable data set. Partitions can be extended to a maximum size of 1, 2, or 4 gigabytes, depending upon the number of partitions in the partitioned page set. All partitions of a given page set have the same maximum size.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. Synonymous with program library.

partitioned page set. A partitioned table space or an index space. Header pages, space map pages, data pages, and index pages reference data only within the scope of the partition.

partitioned table space. A table space subdivided into parts (based upon index key range), each of which may be processed by utilities independently.

partner logical unit. An access point in the SNA network that is connected to the local DB2 by way of a VTAM conversation.

PCT. Program control table (CICS).

PDS. Partitioned data set.

piece. A data set of a nonpartitioned page set.

plan. See *application plan*.

plan allocation. The process of allocating DB2 resources to a plan in preparation to execute it.

plan name. The name of an application plan.

plan segmentation. The dividing of each plan into sections. When a section is needed, it is independently brought into the EDM pool.

PLT. Program list table (CICS).

point of consistency. A time when all recoverable data an application accesses is consistent with other data. Synonymous with *sync point* or *commit point*.

PPT. (1) Processing program table (CICS).
(2) Program properties table (MVS).

precompilation. A processing of application programs containing SQL statements that takes place before compilation. SQL statements are replaced with statements that are recognized by the host language compiler. Output from this precompilation includes source code that can be submitted to the compiler and the database request module (DBRM) that is input to the bind process.

predicate. An element of a search condition that expresses or implies a comparison operation.

prefix. A code at the beginning of a message or record.

primary authorization ID. The authorization ID used to identify the application process to DB2.

primary index. An index that enforces the uniqueness of a primary key.

primary key. A unique, nonnull key that is part of the definition of a table. A table cannot be defined as a parent unless it has a unique key or primary key.

private connection. A communications connection that is specific to DB2.

privilege. The capability of performing a specific function, sometimes on a specific object. The term includes:

explicit privileges, which have names and are held as the result of SQL GRANT and REVOKE statements. For example, the SELECT privilege.

implicit privileges, which accompany the ownership of an object, such as the privilege to drop a synonym one owns, or the holding of an authority, such as the privilege of SYSADM authority to terminate any utility job.

privilege set. For the installation SYSADM ID, the set of all possible privileges. For any other authorization ID, the set of all privileges recorded for that ID in the DB2 catalog.

process. A general term for a unit that depends on the environment, but has the same basic properties in every

program • relational database name (RDBNAM)

environment. A process involves the execution of one or more programs, and is the unit to which resources and locks are allocated. The execution of an SQL statement is always associated with some process.

program. A single compilable collection of executable statements in a programming language.

program temporary fix (PTF). A solution or bypass of a problem diagnosed as a result of a defect in a current unaltered release of a licensed program. An authorized program analysis report (APAR) fix is corrective service for an existing problem. A PTF is preventive service for problems that might be encountered by other users of the product. A PTF is “temporary” because a permanent fix is usually not incorporated into the product until its next release.

protected conversation. A VTAM conversation that supports two-phase commit flows.

PSRCP. Page set recovery pending.

PTF. Program temporary fix.

Q

QMF. Query Management Facility.

QSAM. Queued Sequential Access Method.

R

RACF. OS/VS2 MVS Resource Access Control Facility.

RBA. Relative byte address.

RCT. Resource control table (CICS attachment facility).

RDB. See *relational database*.

RDBMS. Relational database management system.

RDBNAM. See *relational database name*.

RDF. Record definition field.

read stability (RS). An isolation level that is similar to repeatable read but does not completely isolate an application process from all other concurrently executing application processes. Under level RS, an application that issues the same query more than once might read additional rows, known as *phantom rows*, that were inserted and committed by a concurrently executing application process.

rebind. To create a new application plan for an application program that has been bound previously. If,

for example, you have added an index for a table accessed by your application, you must rebind the application in order to take advantage of that index.

record. The storage representation of a row or other data.

record identifier (RID) pool. An area of main storage above the 16MB line that is reserved for sorting record identifiers during list prefetch processing.

recovery. The process of rebuilding databases after a system failure.

recovery log. A collection of records that describes the events that occur during DB2 execution and their sequence. The information recorded is used for recovery in the event of a failure during DB2 execution.

recovery pending (RECP). This condition prevents SQL access to a table space or index space that may need to be recovered.

recovery token. An identifier for an element used in recovery. For example, *NID* or *URID*.

RECP. Recovery pending.

redo. A state of a unit of recovery which indicates that changes made are to be reapplied to the DASD media to ensure data integrity.

referential constraint. The requirement that nonnull values of a designated foreign key are valid only if they equal values of the primary key of a designated table.

referential integrity. The condition that exists when all intended references from data in one column of a table to data in another column of the same or a different table are valid. Maintaining referential integrity requires enforcing referential constraints on all LOAD, RECOVER, INSERT, UPDATE, and DELETE operations.

referential structure. A set of tables and relationships that includes at least one table and, for every table in the set, all the relationships in which that table participates and all the tables to which it is related.

relational database. A database that can be perceived as a set of tables and manipulated in accordance with the relational model of data.

relational database management system (RDBMS). A relational database manager that operates consistently across supported IBM systems.

relational database name (RDBNAM). A unique identifier for an RDBMS within a network. In DB2, this must be the value in the LOCATION column of table SYSIBM.LOCATIONS in the CDB. DB2 publications

refer to the name of another RDBMS as a LOCATION value or a location name.

relationship. A defined connection between the rows of a table or the rows of two tables. A relationship is the internal representation of a referential constraint.

relative byte address (RBA). The offset of a data record or control interval from the beginning of the storage space allocated to the data set or file to which it belongs.

remigration. The process of returning to a current release of DB2 following a fallback to a previous release. This procedure constitutes another migration process.

remote attach request. A request by a remote location to attach to the local DB2 subsystem. Specifically, the request sent is an SNA Function Management Header 5.

remote subsystem. Any RDBMS, except the *local subsystem*, with which the user or application can communicate. The subsystem need not be remote in any physical sense, and may even operate on the same processor under the same MVS system.

repeatable read (RR). The isolation level that provides maximum protection from other executing application programs. When an application program executes with repeatable read protection, rows referenced by the program cannot be changed by other programs until the program reaches a commit point.

request commit. The vote submitted to the prepare phase if the participant has modified data and is prepared to commit or roll back.

requester. Also *application requester (AR)*. The source of a request to a remote RDBMS, the system that requests the data.

request unit (RU). The part of a basic information unit that follows a request header and contains the data.

resource allocation. The part of plan allocation that deals specifically with the database resources.

resource control table (RCT). A construct of the CICS attachment facility, created by site-provided macro parameters, that defines authorization and access attributes for transactions or transaction groups.

resource definition online. A CICS feature that allows you to define CICS resources on line without assembling tables.

resource limit facility (RLF). A portion of DB2 code that prevents dynamic manipulative SQL statements from exceeding specified time limits.

resource limit specification table. A site-defined table that specifies the limits to be enforced by the resource limit facility.

result table. The set of rows specified by a SELECT statement.

RID pool. Record identifier pool.

right outer join. The result of a join operation that includes the matched rows of both tables being joined and preserves the unmatched rows of the second join operand. See also *join*.

RLF. Resource limit facility.

RMID. Resource manager identifier.

RO. Read-only access.

rollback. The process of restoring data changed by SQL statements to the state at its last commit point. All locks are freed. Contrast with *commit*.

root page. The page of an index page set that follows the first index space map page. A root page is the highest level (or the beginning point) of the index.

row. The horizontal component of a table. A row consists of a sequence of values, one for each column of the table.

RRE. Residual recovery entry (IMS).

RRSAF. Recoverable Resource Manager Services attachment facility. A DB2 subcomponent that uses OS/390 Transaction Management and Recoverable Resource Manager Services to coordinate resource commitment between DB2 and all other resource managers that also use OS/390 RRS in an OS/390 system.

RTT. Resource translation table.

RU. Request unit.

S

SBCS. Single-byte character set.

scalar function. An SQL operation that produces a single value from another value and is expressed as a function name followed by a list of arguments enclosed in parentheses. See also *column function*.

search condition. A criterion for selecting rows from a table. A search condition consists of one or more predicates.

secondary authorization ID • SSM

secondary authorization ID. An authorization ID that has been associated with a primary authorization ID by an authorization exit routine.

segmented table space. A table space that is divided into equal-sized groups of pages called segments. Segments are assigned to tables so that rows of different tables are never stored in the same segment.

self-referencing constraint. A referential constraint that defines a relationship in which a table is a dependent of itself.

self-referencing table. A table with a self-referencing constraint.

sequential data set. A non-DB2 data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. Several of the DB2 database utilities require sequential data sets.

server. Also *application server (AS)*. The target for a request from a remote RDBMS, the RDBMS that provides the data.

service request block. A unit of work that is scheduled to execute in another address space.

session. A link between two nodes in a VTAM network.

session protocols. The available set of SNA communication requests and responses.

shared lock. A lock that prevents concurrently executing application processes from changing data, but not from reading data.

short string. A string whose actual length, or a varying-length string whose maximum length, is 255 bytes (127 double-byte characters) or less.

sign-on. A request made on behalf of an individual CICS or IMS application process by an attach facility to enable DB2 to verify that it is authorized to use DB2 resources.

simple page set. A nonpartitioned page set. A simple page set initially consists of a single data set (page set piece). If and when that data set is extended to 2 gigabytes, another data set is created, and so on up to a total of 32 data sets. The data sets are considered by DB2 to be a single contiguous linear address space containing a maximum of 64 gigabytes. Data is stored in the next available location within this address space without regard to any partitioning scheme.

simple table space. A table space that is neither partitioned nor segmented.

single-byte character set (SBCS). A set of characters in which each character is represented by a single byte.

SMF. System management facility.

SMP/E. System Modification Program/Extended.

SMS. Storage Management Subsystem.

SNA. Systems Network Architecture.

SNA network. The part of a network that conforms to the formats and protocols of Systems Network Architecture (SNA).

source program. A set of host language statements and SQL statements that is processed by an SQL precompiler.

special register. A storage area that is defined for a process by DB2 and is used to store information that can be referenced in SQL statements. Examples of special registers are USER, CURRENT DATE, and CURRENT TIME.

SPUFI. SQL Processor Using File Input. A facility of the TSO attachment subcomponent that enables the DB2I user to execute SQL statements without embedding them in an application program.

SQL. Structured Query Language.

SQL authorization ID (SQL ID). The authorization ID that is used for checking dynamic SQL statements in some situations.

SQL communication area (SQLCA). A structure used to provide an application program with information about the execution of its SQL statements.

SQL descriptor area (SQLDA). A structure that describes input variables, output variables, or the columns of a result table.

SQL processing conversation. Any conversation that requires access of DB2 data, either through an application or by dynamic query requests.

SQLCA. SQL communication area.

SQLDA. SQL descriptor area.

SQL/DS. SQL/Data System. Also known as *DB2/VSE & VM*.

SRB. See *service request block*.

SSI. MVS subsystem interface.

SSM. Subsystem member.

stand-alone. An attribute of a program that means it is capable of executing separately from DB2, without using DB2 services.

static SQL. SQL statements, embedded within a program, that are prepared during the program preparation process (before the program is executed). After being prepared, the SQL statement does not change (although values of host variables specified by the statement might change).

storage group. A named set of DASD volumes on which DB2 data can be stored.

string. See *character string* or *graphic string*.

Structured Query Language (SQL). A standardized language for defining and manipulating data in a relational database.

subcomponent. A group of closely related DB2 modules that work together to provide a general function.

subpage. The unit into which a physical index page can be divided.

subquery. A SELECT statement within the WHERE or HAVING clause of another SQL statement; a nested SQL statement.

subselect. That form of a query that does not include ORDER BY clause, UPDATE clause, or UNION operators.

subsystem. A distinct instance of a RDBMS.

sync point. See *commit point*.

synonym. In SQL, an alternative name for a table or view. Synonyms can only be used to refer to objects at the subsystem in which the synonym is defined.

system administrator. The person having the second highest level of authority within DB2. System administrators make decisions about how DB2 is to be used and implement those decisions by choosing system parameters. They monitor the system and change its characteristics to meet changing requirements and new data processing goals.

system agent. A work request that DB2 creates internally.

system conversation. The conversation that two DB2s must establish to process system messages before any distributed processing can begin.

system diagnostic work area (SDWA). The data that is recorded in a SYS1.LOGREC entry that describes a program or hardware error.

System Modification Program/Extended (SMP/E). A tool for making software changes in programming systems (such as DB2 or MVS), and for controlling those changes.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information through and controlling the configuration and operation of networks.

SYS1.DUMPxx data set. A data set that contains a system dump.

SYS1.LOGREC. A service aid that contains important information about program and hardware errors.

T

table. A named data object consisting of a specific number of columns and some number of unordered rows. Synonymous with *base table* or *temporary table*.

table check constraint. A user-defined constraint that specifies the values that specific columns of a base table can contain.

table space. A page set used to store the records in one or more tables.

table space set. A set of table spaces and partitions that should be recovered together because each of them contains a table that is a parent or descendent of a table in one of the others.

task control block (TCB). A control block used to communicate information about tasks within an address space that are connected to DB2. An address space can support many task connections (as many as one per task), but only one address space connection. See *address space connection*.

TCB. MVS task control block.

temporary table. A table created by the SQL CREATE GLOBAL TEMPORARY TABLE statement that is used to hold temporary data. Contrast with *result table* and *temporary table*.

thread. The DB2 structure that describes an application's connection, traces its progress, processes resource functions, and delimits its accessibility to DB2 resources and services. Most DB2 functions execute under a thread structure. See also *allied thread* and *database access thread*.

three-part name. The full name of a table, view, or alias. It consists of a location name, authorization ID, and an object name separated by a period.

time • WTOR

time. A three-part value that designates a time of day in hours, minutes, and seconds.

time duration. A decimal integer that represents a number of hours, minutes, and seconds.

timeout. Abnormal termination of either the DB2 subsystem or of an application because of the unavailability of resources. Installation specifications are set to determine both the amount of time DB2 will wait for IRLM services after starting, and the amount of time IRLM will wait if a resource requested by an application is unavailable. If either of these time specifications is exceeded, a timeout is declared.

time-sharing option (TSO). Provides interactive time sharing from remote terminals.

timestamp. A seven-part value that consists of a date and time expressed in years, months, days, hours, minutes, seconds, and microseconds.

TMP. Terminal Monitor Program.

to-do. A state of a unit of recovery that indicates that the unit of recovery's changes to recoverable DB2 resources are indoubt and must either be applied to the DASD media or backed out, as determined by the commit coordinator.

trace. A DB2 facility that provides the ability to monitor and collect DB2 monitoring, auditing, performance, accounting, statistics, and serviceability (global) data.

TSO. Time-sharing option.

TSO attachment facility. A DB2 facility consisting of the DSN command processor and DB2I. Applications that are not written for the CICS or IMS environments can run under the TSO attachment facility.

U

undo. A state of a unit of recovery that indicates that the changes made by the unit of recovery to recoverable DB2 resources must be backed out.

UNION. An SQL operation that combines the results of two select statements. UNION is often used to merge lists of values obtained from several tables.

unique index. An index which ensures that no identical key values are stored in a table.

uniqueness constraint. The rule that no two values in a primary key or key of a unique index can be the same.

unlock. To release an object or system resource that was previously locked and return it to general availability within DB2.

URE. Unit of recovery element.

URID (unit of recovery ID). The LOGRBA of the first log record for a unit of recovery. The URID also appears in all subsequent log records for that unit of recovery.

UT. Utility-only access.

V

value. The smallest unit of data manipulated in SQL.

varying-length string. A character or graphic string whose length varies within set limits. Contrast with *fixed-length string*.

version. A member of a set of similar programs, DBRMs, or packages.

A version of a program is the source code produced by precompiling the program. The program version is identified by the program name and a timestamp (consistency token).

A version of a DBRM is the DBRM produced by precompiling a program. The DBRM version is identified by the same program name and timestamp as a corresponding program version.

A version of a package is the result of binding a DBRM within a particular database system. The package version is identified by the same program name and consistency token as the DBRM.

view. An alternative representation of data from one or more tables. A view can include all or some of the columns contained in tables on which it is defined.

Virtual Telecommunications Access Method

(VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network.

VSAM. Virtual storage access method.

VTAM. MVS Virtual telecommunication access method.

W

WTO. Write to operator.

WTOR. Write to operator with reply.

X

XRF. Extended recovery facility.

Bibliography

DB2 for OS/390 Version 5

- *Administration Guide*, SC26-8957
- *Application Programming and SQL Guide*, SC26-8958
- *Call Level Interface Guide and Reference*, SC26-8959
- *Command Reference*, SC26-8960
- *Data Sharing: Planning and Administration*, SC26-8961
- *Data Sharing Quick Reference Card*, SX26-3841
- *Diagnosis Guide and Reference*, LY27-9659
- *Diagnostic Quick Reference Card*, LY27-9660
- *Installation Guide*, GC26-8970
- *Application Programming Guide and Reference for Java™*, SC26-9547
- *Licensed Program Specifications*, GC26-8969
- *Messages and Codes*, GC26-8979
- *Reference for Remote DRDA Requesters and Servers*, SC26-8964
- *Reference Summary*, SX26-3842
- *Release Guide*, SC26-8965
- *SQL Reference*, SC26-8966
- *Utility Guide and Reference*, SC26-8967
- *What's New?*, GC26-8971
- *Program Directory*

DB2 PM for OS/390 Version 5

- *Batch User's Guide*, SC26-8991
- *Command Reference*, SC26-8987
- *General Information*, GC26-8982
- *Getting Started on the Workstation*, SC26-8989
- *Master Index*, SC26-8984
- *Messages Manual*, SC26-8988
- *Online Monitor User's Guide*, SC26-8990
- *Report Reference Volume 1*, SC26-8985
- *Report Reference Volume 2*, SC26-8986
- *Program Directory*

Ada/370

- *IBM Ada/370 Language Reference*, SC09-1297
- *IBM Ada/370 Programmer's Guide*, SC09-1414
- *IBM Ada/370 SQL Module Processor for DB2 Database Manager User's Guide*, SC09-1450

APL2

- *APL2 Programming Guide*, SH21-1072
- *APL2 Programming: Language Reference*, SH21-1061
- *APL2 Programming: Using Structured Query Language (SQL)*, SH21-1057

AS/400

- *DB2 for OS/400 SQL Programming*, SC41-4611
- *DB2 for OS/400 SQL Reference*, SC41-4612

BASIC

- *IBM BASIC/MVS Language Reference*, GC26-4026
- *IBM BASIC/MVS Programming Guide*, SC26-4027

C/370

- *IBM SAA AD/Cycle C/370 Programming Guide*, SC09-1356
- *IBM SAA AD/Cycle C/370 Programming Guide for Language Environment/370*, SC09-1840
- *IBM SAA AD/Cycle C/370 User's Guide*, SC09-1763
- *SAA CPI C Reference*, SC09-1308

Character Data Representation Architecture

- # • *Character Data Representation Architecture*
- # • *Overview*, GC09-2207
- # • *Character Data Representation Architecture*
- # • *Reference*, SC09-2190

CICS/ESA

- *CICS/ESA Application Programming Guide*, SC33-1169
- *CICS/ESA Application Programming Reference*, SC33-1170
- *CICS/ESA CICS - RACF Security Guide*, SC33-1185
- *CICS/ESA CICS-Supplied Transactions*, SC33-1168
- *CICS/ESA Customization Guide*, SC33-1165
- *CICS/ESA Data Areas*, LY33-6083
- *CICS/ESA Installation Guide*, SC33-1163
- *CICS/ESA Intercommunication Guide*, SC33-1181
- *CICS/ESA Messages and Codes*, SC33-1177
- *CICS/ESA Operations and Utilities Guide*, SC33-1167
- *CICS/ESA Performance Guide*, SC33-1183
- *CICS/ESA Problem Determination Guide*, SC33-1176
- *CICS/ESA Resource Definition Guide*, SC33-1166
- *CICS/ESA System Definition Guide*, SC33-1164
- *CICS/ESA System Programming Reference*, GC33-1171

CICS/MVS

- *CICS/MVS Application Programming Primer*, SC33-0139
- *CICS/MVS Application Programmer's Reference*, SC33-0512
- *CICS/MVS Facilities and Planning Guide*, SC33-0504
- *CICS/MVS Installation Guide*, SC33-0506
- *CICS/MVS Operations Guide*, SC33-0510
- *CICS/MVS Problem Determination Guide*, SC33-0516
- *CICS/MVS Resource Definition (Macro)*, SC33-0509
- *CICS/MVS Resource Definition (Online)*, SC33-0508

IBM C/C++ for MVS/ESA or OS/390

- *IBM C/C++ for MVS/ESA Library Reference*, SC09-1995
- *IBM C/C++ for MVS/ESA Programming Guide*, SC09-1994
- *IBM C/C++ for OS/390 User's Guide*, SC09-2361

IBM COBOL for MVS & VM

- *IBM COBOL for MVS & VM Language Reference*, SC26-4769
- *IBM COBOL for MVS & VM Programming Guide*, SC26-4767

Conversion Guides

- *DBMS Conversion Guide: DATACOM/DB to DB2*, GH20-7564
- *DBMS Conversion Guide: IDMS to DB2*, GH20-7562
- *DBMS Conversion Guide: Model 204 to DB2 or SQL/DS*, GH20-7565
- *DBMS Conversion Guide: VSAM to DB2*, GH20-7566
- *IMS-DB and DB2 Migration and Coexistence Guide*, GH21-1083

Cooperative Development Environment

- *CoOperative Development Environment/370: Debug Tool*, SC09-1623

DATABASE 2 for Common Servers

- *DATABASE 2 Administration Guide for common servers*, S20H-4580
- *DATABASE 2 Application Programming Guide for common servers*, S20H-4643
- *DATABASE 2 Software Developer's Kit for AIX: Building Your Applications*, S20H-4780
- *DATABASE 2 Software Developer's Kit for OS/2: Building Your Applications*, S20H-4787
- *DATABASE 2 SQL Reference for common servers*, S20H-4665
- *DATABASE 2 Call Level Interface Guide and Reference for common servers*, S20H-4644

Data Extract (DXT)

- *Data Extract Version 2: General Information*, GC26-4666
- *Data Extract Version 2: Planning and Administration Guide*, SC26-4631

DataPropagator NonRelational

- *DataPropagator NonRelational MVS/ESA Administration Guide*, SH19-5036
- *DataPropagator NonRelational MVS/ESA Reference*, SH19-5039

DataPropagator Relational

- *DataPropagator Relational User's Guide*, SC26-3399
- *IBM An Introduction to DataPropagator Relational*, GC26-3398

Data Facility Data Set Services

- *Data Facility Data Set Services: User's Guide and Reference*, SC26-4125

Database Design

- *DB2 Database Design and Implementation Using DB2*, SH24-6101
- *DB2 Design and Development Guide*, Gabrielle Wiorkowski and David Kull, Addison Wesley
- *Handbook of Relational Database Design*, C. Fleming and B Von Halle, Addison Wesley
- *Principles of Database Systems*, Jeffrey D. Ullman, Computer Science Press

DataHub

- *IBM DataHub General Information*, GC26-4874

DB2 Universal Database

- *DB2 Universal Database Administration Guide*, S10J-8157
- *DB2 Universal Database API Reference*, S10J-8167
- *DB2 Universal Database Application Development Guide*, SC09-2845
- *DB2 Universal Database Building Applications for UNIX Environments*, S10J-8161
- *DB2 Universal Database Building Applications for Windows and OS/2 Environments*, S10J-8160
- *DB2 Universal Database CLI Guide and Reference*, S10J-8159
- *DB2 Universal Database SQL Reference*, S10J-8165

Device Support Facilities

- *Device Support Facilities User's Guide and Reference*, GC35-0033

DFSMS/MVS

- *DFSMS/MVS: Access Method Services for the Integrated Catalog*, SC26-4906
- *DFSMS/MVS: Access Method Services for VSAM Catalogs*, SC26-4905
- *DFSMS/MVS: Administration Reference for DFSMSdss*, SC26-4929
- *DFSMS/MVS: DFSMSHsm Managing Your Own Data*, SH21-1077
- *DFSMS/MVS: Diagnosis Reference for DFSMSdfp*, LY27-9606
- *DFSMS/MVS: Macro Instructions for Data Sets*, SC26-4913
- *DFSMS/MVS: Managing Catalogs*, SC26-4914
- *DFSMS/MVS: Program Management*, SC26-4916
- *DFSMS/MVS: Storage Administration Reference for DFSMSdfp*, SC26-4920
- *DFSMS/MVS: Using Advanced Services for Data Sets*, SC26-4921
- *DFSMS/MVS: Utilities*, SC26-4926
- *MVS/DFP: Managing Non-VSAM Data Sets*, SC26-4557

DFSORT

- *DFSORT Application Programming: Guide*, SC33-4035

Distributed Relational Database

- *Data Stream and OPA Reference*, SC31-6806
- *Distributed Relational Database Architecture: Application Programming Guide*, SC26-4773
- *Distributed Relational Database Architecture: Connectivity Guide*, SC26-4783
- *Distributed Relational Database Architecture: Evaluation and Planning Guide*, SC26-4650
- *Distributed Relational Database Architecture: Problem Determination Guide*, SC26-4782
- *Distributed Relational Database: Every Manager's Guide*, GC26-3195
- *IBM SQL Reference*, SC26-8416
- *Open Group Technical Standard (the Open Group presently makes the following books available through their website at www.opengroup.org):*
 - *DRDA Volume 1: Distributed Relational Database Architecture (DRDA)*, ISBN 1-85912-295-7
 - *DRDA Volume 3: Distributed Database Management (DDM) Architecture*, ISBN 1-85912-206-X

Education

- *Dictionary of Computing*, SC20-1699
- *IBM Enterprise Systems Training Solutions Catalog*, GR28-5467

Enterprise System/9000 and Enterprise System/3090

- *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide*, GA22-7123

FORTRAN

- *VS FORTRAN Version 2: Language and Library Reference*, SC26-4221
- *VS FORTRAN Version 2: Programming Guide for CMS and MVS*, SC26-4222

High Level Assembler

- *High Level Assembler/MVS and VM and VSE Language Reference*, SC26-4940
- *High Level Assembler/MVS and VM and VSE Programmer's Guide*, SC26-4941

Parallel Sysplex Library

- *System/390 MVS Sysplex Application Migration*, GC28-1211
- *System/390 MVS Sysplex Hardware and Software Migration*, GC28-1210
- *System/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism*, GC28-1208
- *System/390 MVS Sysplex Systems Management*, GC28-1209
- *System/390 MVS 9672/9674 System Overview*, GA22-7148

ICSF/MVS

- *ICSF/MVS General Information*, GC23-0093

IMS/ESA

- *IMS Batch Terminal Simulator General Information*, GH20-5522
- *IMS/ESA Administration Guide: System*, SC26-8013
- *IMS/ESA Application Programming: Database Manager*, SC26-8727
- *IMS/ESA Application Programming: Design Guide*, SC26-8016
- *IMS/ESA Application Programming: Transaction Manager*, SC26-8729
- *IMS/ESA Customization Guide*, SC26-8020
- *IMS/ESA Installation Volume 1: Installation and Verification*, SC26-8023
- *IMS/ESA Installation Volume 2: System Definition and Tailoring*, SC26-8024
- *IMS/ESA Messages and Codes*, SC26-8028
- *IMS/ESA Operator's Reference*, SC26-8030
- *IMS/ESA Utilities Reference: System*, SC26-8035

ISPF

- *ISPF Version 4 Messages and Codes*, SC34-4450
- *ISPF Version 4 for MVS Dialog Management Guide*, SC34-4213
- *ISPF/PDF Version 4 for MVS Guide and Reference*, SC34-4258

- *ISPF and ISPF/PDF Version 4 for MVS Planning and Customization*, SC34-4134

Language Environment for MVS & VM

- *Language Environment for MVS & VM Concepts Guide*, GC26-4786
- *Language Environment for MVS & VM Debugging and Run-Time Messages Guide*, SC26-4829
- *Language Environment for MVS & VM Installation and Customization*, SC26-4817
- *Language Environment for MVS & VM Programming Guide*, SC26-4818
- *Language Environment for MVS & VM Programming Reference*, SC26-3312

MVS/ESA

- *MVS/ESA Analyzing Resource Measurement Facility Monitor I and Monitor II Reference and User's Guide*, LY28-1007
- *MVS/ESA Analyzing Resource Measurement Facility Monitor III Reference and User's Guide*, LY28-1008
- *MVS/ESA Application Development Reference: Assembler Callable Services for OpenEdition MVS*, SC23-3020
- *MVS/ESA Data Administration: Utilities*, SC26-4516
- *MVS/ESA Diagnosis: Procedures*, LY28-1844
- *MVS/ESA Diagnosis: Tools and Service Aids*, LY28-1845
- *MVS/ESA Initialization and Tuning Guide*, SC28-1451
- *MVS/ESA Initialization and Tuning Reference*, SC28-1452
- *MVS/ESA Installation Exits*, SC28-1459
- *MVS/ESA JCL Reference*, GC28-1479
- *MVS/ESA JCL User's Guide*, GC28-1473
- *MVS/ESA JES2 Initialization and Tuning Guide*, SC28-1453
- *MVS/ESA MVS Configuration Program*, GC28-1615
- *MVS/ESA Planning: Global Resource Serialization*, GC28-1450
- *MVS/ESA Planning: Operations*, GC28-1441
- *MVS/ESA Planning: Workload Management*, GC28-1493
- *MVS/ESA Programming: Assembler Services Guide*, GC28-1466
- *MVS/ESA Programming: Assembler Services Reference*, GC28-1474
- *MVS/ESA Programming: Authorized Assembler Services Guide*, GC28-1467
- *MVS/ESA Programming: Authorized Assembler Services Reference, Volumes 1-4*, GC28-1475, GC28-1476, GC28-1477, GC28-1478
- *MVS/ESA Programming: Extended Addressability Guide*, GC28-1468
- *MVS/ESA Programming: Sysplex Services Guide*, GC28-1495
- *MVS/ESA Programming: Sysplex Services Reference*, GC28-1496

- *MVS/ESA Programming: Workload Management Services*, GC28-1494
- *MVS/ESA Routing and Descriptor Codes*, GC28-1487
- *MVS/ESA Setting Up a Sysplex*, GC28-1449
- *MVS/ESA SPL: Application Development Guide*, GC28-1852
- *MVS/ESA System Codes*, GC28-1486
- *MVS/ESA System Commands*, GC28-1442
- *MVS/ESA System Management Facilities (SMF)*, GC28-1457
- *MVS/ESA System Messages Volume 1*, GC28-1480
- *MVS/ESA System Messages Volume 2*, GC28-1481
- *MVS/ESA System Messages Volume 3*, GC28-1482
- *MVS/ESA Using the Subsystem Interface*, SC28-1502

Net.Data for OS/390

- # • *Net.Data Language Environment Guide*,
<http://www.ibm.com/software/net.data/docs>
- # • *Net.Data Programming Guide*,
<http://www.ibm.com/software/net.data/docs>
- # • *Net.Data Reference Guide*,
<http://www.ibm.com/software/net.data/docs>

NetView

- *NetView Installation and Administration Guide*, SC31-8043
- *NetView User's Guide*, SC31-8056

ODBC

- *ODBC 2.0 Programmer's Reference and SDK Guide*, ISBN 1-55615-658-8
- *Inside ODBC*, ISBN 1-55615-815-7

OS/390

- *OS/390 C/C++ Programming Guide*, SC09-2362
- *OS/390 C/C++ Run-Time Library Reference*, SC28-1663
- *OS/390 Information Roadmap*, GC28-1727
- *OS/390 Introduction and Release Guide*, GC28-1725
- *OS/390 JES2 Initialization and Tuning Guide*, SC28-1791
- *OS/390 JES3 Initialization and Tuning Guide*, SC28-1802
- *OS/390 Language Environment for OS/390 & VM Concepts Guide*, GC28-1945
- *OS/390 Language Environment for OS/390 & VM Customization*, SC28-1941
- *OS/390 Language Environment for OS/390 & VM Debugging Guide*, SC28-1942
- *OS/390 Language Environment for OS/390 & VM Programming Guide*, SC28-1939
- *OS/390 Language Environment for OS/390 & VM Programming Reference*, SC28-1940
- *OS/390 MVS Diagnosis: Procedures*, LY28-1082
- *OS/390 MVS Diagnosis: Reference*, SY28-1084

- OS/390 MVS *Diagnosis: Tools and Service Aids*, LY28-1085
- OS/390 MVS *Initialization and Tuning Guide*, SC28-1751
- OS/390 MVS *Initialization and Tuning Reference*, SC28-1752
- OS/390 MVS *Installation Exits*, SC28-1753
- OS/390 MVS *JCL Reference*, GC28-1757
- OS/390 MVS *JCL User's Guide*, GC28-1758
- OS/390 MVS *Planning: Global Resource Serialization*, GC28-1759
- OS/390 MVS *Planning: Operations*, GC28-1760
- OS/390 MVS *Planning: Workload Management*, GC28-1761
- OS/390 MVS *Programming: Assembler Services Guide*, GC28-1762
- OS/390 MVS *Programming: Assembler Services Reference*, GC28-1910
- OS/390 MVS *Programming: Authorized Assembler Services Guide*, GC28-1763
- OS/390 MVS *Programming: Authorized Assembler Services Reference, Volumes 1-4*, GC28-1764, GC28-1765, GC28-1766, GC28-1767
- OS/390 MVS *Programming: Callable Services for High-Level Languages*, GC28-1768
- OS/390 MVS *Programming: Extended Addressability Guide*, GC28-1769
- OS/390 MVS *Programming: Sysplex Services Guide*, GC28-1771
- OS/390 MVS *Programming: Sysplex Services Reference*, GC28-1772
- OS/390 MVS *Programming: Workload Management Services*, GC28-1773
- OS/390 MVS *Routing and Descriptor Codes*, GC28-1778
- OS/390 MVS *Setting Up a Sysplex*, GC28-1779
- OS/390 MVS *System Codes*, GC28-1780
- OS/390 MVS *System Commands*, GC28-1781
- OS/390 MVS *System Messages Volume 1*, GC28-1784
- OS/390 MVS *System Messages Volume 2*, GC28-1785
- OS/390 MVS *System Messages Volume 3*, GC28-1786
- OS/390 MVS *System Messages Volume 4*, GC28-1787
- OS/390 MVS *System Messages Volume 5*, GC28-1788
- OS/390 *Security Server (RACF) Auditor's Guide*, SC28-1916
- OS/390 *Security Server (RACF) Command Language Reference*, SC28-1919
- OS/390 *Security Server (RACF) General User's Guide*, SC28-1917
- OS/390 *Security Server (RACF) Security Administrator's Guide*, SC28-1915
- OS/390 *Security Server (RACF) System Programmer's Guide*, SC28-1913
- OS/390 *SMP/E Reference*, SC28-1806

- OS/390 *SMP/E User's Guide*, SC28-1740
- OS/390 *RMF User's Guide*, SC28-1949
- OS/390 *TSO/E CLISTS*, SC28-1973
- OS/390 *TSO/E Command Reference*, SC28-1969
- OS/390 *TSO/E Customization*, SC28-1965
- OS/390 *TSO/E Messages*, GC28-1978
- OS/390 *TSO/E Programming Guide*, SC28-1970
- OS/390 *TSO/E Programming Services*, SC28-1971
- OS/390 *TSO/E REXX Reference*, SC28-1975
- OS/390 *TSO/E User's Guide*, SC28-1968

OS/390 OpenEdition

- OS/390 *OpenEdition DCE Administration Guide*, SC28-1584
- OS/390 *OpenEdition DCE Introduction*, GC28-1581
- OS/390 *R1 OE DCE Messages and Codes*, ST01-0920
- OS/390 *OpenEdition Command Reference*, SC28-1892
- OS/390 *OpenEdition Messages and Codes*, SC28-1908
- OS/390 *OpenEdition Planning*, SC28-1890
- OS/390 *OpenEdition User's Guide*, SC28-1891

PL/I for MVS & VM

- *IBM PL/I MVS & VM Language Reference*, SC26-3114
- *IBM PL/I MVS & VM Programming Guide*, SC26-3113

OS PL/I

- OS *PL/I Programming Language Reference*, SC26-4308
- OS *PL/I Programming Guide*, SC26-4307

PROLOG

- *IBM SAA AD/Cycle Prolog/MVS & VM Programmer's Guide*, SH19-6892

Query Management Facility

- *Query Management Facility: Managing QMF for MVS*, SC26-8218
- *Query Management Facility: Reference*, SC26-4716
- *Query Management Facility: Using QMF*, SC26-8078

Remote Recovery Data Facility

- *Remote Recovery Data Facility Program Description and Operations*, LY37-3710

Resource Access Control Facility (RACF)

- *External Security Interface (RACROUTE) Macro Reference for MVS and VM*, GC28-1366
- *Resource Access Control Facility (RACF) Auditor's Guide*, SC28-1342
- *Resource Access Control Facility (RACF) Command Language Reference*, SC28-0733

- *Resource Access Control Facility (RACF) General Information Manual, GC28-0722*
- *Resource Access Control Facility (RACF) General User's Guide, SC28-1341*
- *Resource Access Control Facility (RACF) Security Administrator's Guide, SC28-1340*
- *Resource Access Control Facility (RACF) System Programmer's Guide, SC28-1343*

Storage Management

- *MVS/ESA Storage Management Library: Implementing System-Managed Storage, SC26-3123*
- *MVS/ESA Storage Management Library: Leading an Effective Storage Administration Group, SC26-3126*
- *MVS/ESA Storage Management Library: Managing Data, SC26-3124*
- *MVS/ESA Storage Management Library: Managing Storage Groups, SC26-3125*
- *MVS Storage Management Library: Storage Management Subsystem Migration Planning Guide, SC26-4659*

System/370 and System/390

- *IBM System/370 ESA Principles of Operation, SA22-7200*
- *IBM System/390 ESA Principles of Operation, SA22-7205*
- *System/390 MVS Sysplex Hardware and Software Migration, GC28-1210*

System Modification Program Extended (SMP/E)

- *System Modification Program Extended (SMP/E) Reference, SC28-1107*
- *System Modification Program Extended (SMP/E) User's Guide, SC28-1302*

System Network Architecture (SNA)

- *SNA Formats, GA27-3136*
- *SNA LU 6.2 Peer Protocols Reference, SC31-6808*
- *SNA Transaction Programmer's Reference Manual for LU Type 6.2, GC30-3084*
- *SNA/Management Services Alert Implementation Guide, GC31-6809*

TCP/IP

- *IBM TCP/IP for MVS: Customization & Administration Guide, SC31-7134*
- *IBM TCP/IP for MVS: Diagnosis Guide, LY43-0105*
- *IBM TCP/IP for MVS: Messages and Codes, SC31-7132*
- *IBM TCP/IP for MVS: Planning and Migration Guide, SC31-7189*

TSO Extensions

- *TSO/E CLISTS, SC28-1876*
- *TSO/E Command Reference, SC28-1881*
- *TSO/E Customization, SC28-1872*
- *TSO/E Messages, GC28-1885*
- *TSO/E Programming Guide, SC28-1874*
- *TSO/E Programming Services, SC28-1875*
- *TSO/E User's Guide, SC28-1880*

VS COBOL II

- *VS COBOL II Application Programming Guide for MVS and CMS, SC26-4045*
- *VS COBOL II Application Programming: Language Reference, SC26-4047*
- *VS COBOL II Installation and Customization for MVS, SC26-4048*

VTAM

- *Planning for NetView, NCP, and VTAM, SC31-8063*
- *VTAM for MVS/ESA Diagnosis, LY43-0069*
- *VTAM for MVS/ESA Messages and Codes, SC31-6546*
- *VTAM for MVS/ESA Network Implementation Guide, SC31-6548*
- *VTAM for MVS/ESA Operation, SC31-6549*
- *VTAM for MVS/ESA Programming, SC31-6550*
- *VTAM for MVS/ESA Programming for LU 6.2, SC31-6551*
- *VTAM for MVS/ESA Resource Definition Reference, SC31-6552*

Index

Numerics

32KB buffering 2-252

4KB buffering 2-252

A

ACBNAME option of APPL statement 3-17

ACCEPT job 2-71, 2-84

active log

data set

DASD requirements 2-18

prefix 2-114

shared read-only data 2-263

description 1-28

dual logging 2-112

installation 2-239

size

estimating 2-18, 2-139

address space

allied agent 2-36

database services

description 2-35

working storage calculation 2-46, 2-47

DB2 1-33

DDF 2-35

DSN1DIST 2-35

IRLM 2-35

system services 2-35

ADSNDKF library 2-70

ADSNENU library 2-70

ADSNLOAD library 2-70

ADSNMACS library 2-70

ADXRLoad library 2-70

allied agent address space 2-36

allocating storage

dictionary 2-28

table 2-27

allocation job 2-80

ALLOCATION UNITS

field of panel DSNTIPA 2-193

AMDPRECT module 2-234

APN option of DSNMAPN macro 2-376, 2-378

APPC option of APPL statement 3-16

APPL REGISTRATION TABLE

field of panel DSNTIPZ 2-211

APPL statement

description 3-14

example 3-14

option descriptions 3-14, 3-18

options

DSESLIM 3-37

LUNAME 3-46

APPLICATION LOAD

field of panel DSNTIPT 2-118

application program

coding SQL statements

data communication coding 1-39

performance X-59

APPLY job 2-83

archive log

cataloging options 2-106

data set

DASD requirements 2-24

prefix 2-115

shared read-only data 2-263

description 1-28

dual logging 2-112

ARCHIVE LOG FREQ

field of panel DSNTIPL 2-191

ARCHIVE LOG PW

field of panel DSNTIPP 2-182

ARCHIVE LOG RACF

field of panel DSNTIPP 2-182

ART/ORT ESCAPE CHARACTER

field of panel DSNTIPZ 2-210

ASCII

conversion table X-56

ASCII CODED CHARACTER SET

field of panel DSNTIPF 2-169

ASSEMBLER

field of panel DSNTIPG 2-130

ASSISTANT

field of panel DSNTIPK 2-111

Asynchronous Transfer Mode (ATM) 3-6

See also ATM support

ATCSTRxx member of SYS1.VTAMLST library 3-26

ATM support 3-6

ATNLOSS option of APPL statement 3-16

attachment facility

CICS 2-250, 2-278

See also CICS

IMS 2-371

See also IMS

installation 2-250, 2-371

migration 2-278

TSO 2-243, 2-275

See also TSO

AUDIT TRACE

field of panel DSNTIPN 2-154

AUTH MEMBER

field of panel DSNTIPM 2-188

AUTH option

APPL statement 3-16

DSNCRCT macro

TYPE=ENTRY 2-409

AUTH SEQUENCE
 field of panel DSNTIPM 2-189
 AUTHID
 column of MODESELECT table 3-31
 column of USERNAMES table 3-23, 3-58
 authorization ID
 RACF 2-250
 TSO 2-250
 AUTO BIND
 field of panel DSNTIPO 2-160, 2-226, 2-304
 AUTO START
 field of panel DSNTIPI 2-175
 automatic
 recall 2-158
 AUTOSSES option of APPL statement 3-14
 availability
 improved 1-32

B

binding
 installation 2-254
 migration
 DCLGEN 2-287
 SPUFI 2-287
 remote package
 plan name for 3-32
 relation to character conversion X-64
 bit data
 columns X-59
 BLOCK SIZE
 field of panel DSNTIPA 2-195
 bootstrap data set (BSDS) 2-281
See also BSDS (bootstrap data set)
 BP0-BP29 fields of panel DSNTIP1 2-150
 BP30-BP49 fields of panel DSNTIP2 2-152
 BP32K-BP32K9 fields of panel DSNTIP2 2-152
 BSDS (bootstrap data set)
 adding a second 2-281
 description 1-29
 installation 2-239
 space requirement 2-21
 BSDS PASSWORD
 field of panel DSNTIPP 2-182
 buffer pool
 description 1-29
 storage requirement 2-38
 VTAM IOBUF storage requirements 3-36

C

C fields of panel DSNTIPU 2-123, 2-124
 C++ fields of panel DSNTIPU 2-126
 CAF (call attachment facility)
 description 1-38

call attachment facility (CAF) 1-38
See also CAF (call attachment facility)
 capturing changed data
 log data set size 2-21
 CATALOG ALIAS
 field of panel DSNTIPA2 2-105
 catalog columns
 revised 2-270, 2-300
 CATALOG DATA
 field of panel DSNTIPA 2-194
 catalog tables
 SYSSTRINGS
 description X-62
 governing rules X-64
 catalog, DB2
 DASD requirements 2-18
 description 1-26
 installing 2-239
 catalog, integrated catalog facility 2-238
See also integrated catalog facility
 CCSID (coded character set identifier)
 changing X-55
 code page X-59
 definition X-56
 installation option for 2-168
 specifying X-56
 CD-ROM, books on 1-7
 CDB (communications database)
 creating
 during installation 2-254, 2-255
 description 1-30, 3-20, 3-56
 dropping while DDF is active 3-32
 LOCATIONS table of CDB'.tables 3-3
See also SYSIBM
 LUMODES table of CDB'.tables 3-3
See also SYSIBM
 LUNAMES table of CDB'.tables 3-3
See also SYSIBM
 MODESELECT table of CDB'.tables 3-3
See also SYSIBM
 populating while
 connecting DB2 subsystems 3-20, 3-56
 installing 2-256
 updating while DDF is active 3-32
 USERNAMES table of CDB'.tables 3-3
See also SYSIBM
 change log inventory utility
 changing
 generic LU name 3-46
 location name 3-46
 LU name 3-46
 VTAM password 3-46
 changed data capture 2-21
See also capturing changed data
 channel-to-channel (CTC) 3-42
See also CTC (channel-to-channel)

- character conversion
 - adding a conversion procedure X-63
 - code page X-59
 - code points X-59
 - description X-55
 - distributed data X-55
 - expanding conversion X-60
 - installation procedure X-59
 - performance X-59
 - rebinding remote packages X-64
 - SYSIBM.SYSSTRINGS catalog table X-62
- character string
 - transmitting X-55
- checkpoint
 - frequency 2-21
- CHECKPOINT FREQ
 - field of panel DSNTIPN 2-157
- CICS
 - attachment facility
 - connecting 2-379
 - commands
 - DSNCCOM1 transaction program 2-395
 - connecting to DB2
 - accessing resources 2-397
 - authorization 2-396
 - authorization IDs 2-402
 - CICS V3 and earlier 2-380
 - CICS V4 and subsequent releases 2-379
 - installation 2-250
 - security 2-396
 - description, attachment facility 1-38
 - facilities
 - message destination 2-400
 - processing program table 2-393
 - program control table 2-389
 - resource control table 2-385
 - subtask failure snap dump 2-403
 - initialization JCL 2-396, 2-404
 - migration 2-278, 2-379
 - operating
 - create thread error processing 2-400
 - recovery from system failure 1-39
 - starting 2-403
 - starting an application 2-330
 - testing 2-327
 - using 2-330
 - planning
 - DB2 considerations 1-38
 - OSCOR storage area 2-386
 - storage requirements 2-381
 - system administration 1-40
 - system tables
 - program entries 2-390
 - program list table 2-394
 - system initialization table 2-386
 - transaction entries 2-387
- CICS (*continued*)
 - table generation procedures 2-398
 - thread
 - maximum, specifying 2-404, 2-406
 - use of XRF (extended recovery facility) 1-39
- CICS attachment facility 2-379
 - See *also* CICS
- CICS fields of panel DSNTIP3 2-136
- CICS RELEASE
 - field of panel DSNTIPV 2-135
- class of service, modifying 3-28
- CLIST LIBRARY
 - field of panel DSNTIPT 2-118
- CLIST messages fields of panel DSNTIPC1 2-219
- clustering index
 - description 1-25
- CNOS (change number of sessions)
 - description 3-37
- COBOL TYPE
 - field of panel DSNTIPY 2-213
- code page X-59
- coded character set X-55
- CODED CHARACTER SET fields of panel DSNTIPF 2-168
 - See *also* CCSID (coded character set identifier)
- column
 - description 1-19, 1-24
- COLUMNS
 - field of panel DSNTIPD 2-140
- command recognition character (CRC) 2-373
 - See *also* CRC (command recognition character)
- common service area (CSA) 2-24
 - See *also* CSA (common service area)
- communications database (CDB) 3-3, 3-22
 - See *also* CDB (communications database)
- COMPACT DATA
 - field of panel DSNTIPA 2-198
- compression
 - archive log data set 2-198
- COMPROT option of MODEENT macro
 - description 3-20
- connection
 - database management systems 3-3, 3-21, 3-57
 - DB2 3-21, 3-57
 - systems 3-3, 3-6
 - DB2I panels to ISPF
 - installation 2-246
 - migration 2-276
- connection exit routine
 - installation 2-241
 - migration 2-284
- contention
 - for conversation 3-14
- continuous operation
 - recovery planning 1-32

CONTRACT THREAD STG
 field of panel DSNTIPE 2-226
 subsystem parameter
 CONSTOR 2-226
 CONTROL ALL APPLICATIONS
 field of panel DSNTIPZ 2-210
 conversation
 contention 3-14
 definition 3-9
 SQL 3-30
 See also SQL processing conversation
 system 3-22
 See also system conversation
 conversion procedure
 adding X-63
 conversion table X-55
 rules for X-64
 CONVLIMIT
 column of LUMODES table
 CNOS negotiation 3-37
 description 3-31
 COORDINATOR
 field of panel DSNTIPK 2-111
 COPY 1 NAME
 field of panel DSNTIPH 2-113
 COPY 1 PREFIX
 field of panel DSNTIPH
 active log 2-114
 archive log 2-115
 COPY 2 NAME
 field of panel DSNTIPH 2-113
 COPY 2 PREFIX
 field of panel DSNTIPH
 active log 2-114
 archive log 2-115
 CRC (command recognition character)
 installation 2-373
 CROSS MEMORY
 field of panel DSNTIPJ 2-177
 CSA (common service area) 2-24, 2-36
 CSD NAME
 field of panel DSNTIP3 2-137
 CTC (channel-to-channel)
 considerations for MAXBFRU size 3-42
 sample definitions for VTAM 3-42
 SNA X'800A' sense code 3-42
 Customer Information Control System (CICS) 1-40
 See also CICS

D

DASD
 requirements 2-16, 2-25
 storage group 1-22
 data
 availability after I/O error 1-32

data (*continued*)
 compression 2-363
 distributed 3-3
 See also distributed data
 data compression
 dictionary
 description 2-28
 estimating DASD storage 2-28
 estimating virtual storage 2-29
 Huffman 2-363
 Data Facility Sort (DFSORT) 2-236
 See also DFSORT (Data Facility Sort)
 Data Propagator NonRelational (DPropNR) 1-41
 See also DPropNR (Data Propagator
 data set
 control block size calculation 2-45
 storage requirements 2-16
 DATA SET NAME(MEMBER)
 field of panel DSNTIPA1 2-101
 DATA SHARING
 field of panel DSNTIPA1 2-100
 data structure
 hierarchy 1-21
 types 1-21
 database
 default database 1-22
 See also default database (DSNDB04)
 description 1-22
 DSNDB06 (DB2 catalog database) 2-239
 See also DSNDB06 database
 DSNDB07 (work file database) 2-252
 See also work file database
 relational 1-19
 temporary storage estimation 2-21
 database descriptor (DBD) 1-27
 See also DBD (database descriptor)
 database services address space 2-35
 DATABASES field
 panel DSNTIPD 2-140
 panel DSNTIPE 2-145
 DATE
 option of DSNHDECP macro 2-170
 DATE FORMAT
 field of panel DSNTIP4 2-170
 DB2 books on line 1-7
 DB2 Connect
 creating database
 during installation 2-255
 storage estimation 2-24
 DB2 GENERIC LUNAME
 field of panel DSNTIPR 2-203, 3-12
 DB2 Interactive (DB2I) 1-31
 See also DB2I (DB2 Interactive)
 DB2 LOCATION NAME
 field of panel DSNTIPR 2-202

DB2 NETWORK LUNAME
 field of panel DSNTIPR 2-202

DB2 NETWORK PASSWORD
 field of panel DSNTIPR 2-202

DB2 Online Help
 accessing 2-65
 adding SEOYLOAD to the LINKLIST 2-57
 adding SEOYLPA to the PLPA or the
 LINKLIST 2-57
 allocating data sets 2-56
 APPLY step 2-56
 changing book data set names 2-61
 copying sample installation jobs 2-54
 copying the online dictionary 2-57
 customizing BookManager READ/MVS 2-61
 EOXVOPTS member 2-57
 exiting 2-67
 installation 2-51
 LINK-EDIT step 2-57
 maintaining 2-67
 making available to users 2-58
 performing the SMP/E receive step 2-56
 reading 2-65
 searching text 2-65
 storage requirements 2-52
 unloading DB2 books 2-59
 using 2-65

DB2 private protocol access
 definition 3-3
 setting up 3-3

DB2 PROC NAME
 field of panel DSNTIPX 2-207

DB2 SAMPLES GROUP
 field of panel DSNTIP3 2-137

DB2I (DB2 Interactive)
 connecting to ISPF
 installation 2-246
 migration 2-276
 description 1-31
 English panels, fall back 2-87
 Kanji panels, fall back 2-87
 panels
 description 1-37

DBCS (double-byte character set)
 identifiers X-58

DBD (database descriptor)
 contents 1-27
 size 2-44

DBD01 directory table space
 contents 1-27

DBRM LIBRARY
 field of panel DSNTIPT 2-118

DBRMLIB.DATA library data set
 DASD volume 2-108
 device type 2-109
 DSNTIJIN job 2-239

DBRMLIB.DATA library data set (*continued*)
 installing a second DB2 2-260, 2-262
 naming considerations 2-78

DCLGEN (declarations generator)
 installation 2-320
 migration 2-287

DDCS (data definition control support)
 creating database
 during installation 2-255
 database 1-30
 storage estimation 2-24

DDF
 overview 3-4

DDF (distributed data facility)
 address space 2-35
 description 1-42

DDF STARTUP OPTION
 field of panel DSNTIPR 2-201

DDRAINL option of APPL statement 3-17

deadlock
 cycles 2-179
 sync point rollback 2-402

DEADLOCK CYCLE
 field of panel DSNTIPJ 2-179

DEADLOCK TIME
 field of panel DSNTIPJ 2-179

DEALLOC PERIOD
 field of panel DSNTIPA 2-195

DECIMAL ARITHMETIC
 field of panel DSNTIP4 2-172

DECIMAL POINT IS
 field of panel DSNTIPF 2-165

DECLARATION LIBRARY
 field of panel DSNTIPT 2-118

DEF ENCODING SCHEME
 field of panel DSNTIPF 2-169

default database (DSNDB04)
 defining 1-22
 storage estimation 1-22, 2-23

DEFAULT INDEX TYPE
 field of panel DSNTIPE 2-148

DEFER
 field of panel DSNTIPS 2-200

DEFINE CATALOG
 field of panel DSNTIPA2 2-107

DESCSTAT
 parameter of DSNTIJUZ 3-4

DEVICE TYPE 1
 field of panel DSNTIPA
 description 2-194

DEVICE TYPE 2
 field of panel DSNTIPA
 description 2-194

DFHSIP statement 2-404

DFSESL DD statement 2-76, 2-371

DFSMS (Data Facility Storage Management Subsystem)
 data set passwords 2-181
 description 1-36
 DFSMSdfp partitioned data set extended (PDSE) 1-36
 DFSMSHsm (Data Facility Hierarchical Storage Manager)
 RECALL command 2-158
 DFSORT (Data Facility Sort)
 program library 2-236
 dictionary 2-28
 See *also* data compression
 directory
 DASD requirements 2-18
 description 1-27
 installing 2-239
 panels 2-92
 table space names 1-27
 DIRECTORY/CATALOG
 field of panel DSNTIPP 2-183
 DISCONNECT IRLM
 field of panel DSNTIPJ 2-226
 DISPLAY NET,BFRUSE command of VTAM 3-25
 DIST SQL STR DELIMTR
 field of panel DSNTIPF 2-167
 distributed data
 planning
 DB2 private protocol access 3-3
 DRDA access 3-3
 number of systems that can be connected 3-3
 programming
 character conversion X-55
 testing 2-333
 distributed data facility (DDF) 1-42
 See *also* DDF (distributed data facility)
 Distributed Relational Database Architecture (DRDA) 3-4
 See *also* DRDA (Distributed Relational Database Architecture)
 distributed unit of work 3-3
 See *also* DB2 private protocol access
 distribution libraries 2-70
 DASD requirements 2-16
 manage using DFSMSHsm 2-158
 DL/I
 batch
 features 1-42
 DMINWNL option of APPL statement 3-14
 DMINWNR option of APPL statement 3-15
 domain name
 definition 3-49
 domain name server
 definition 3-49
 DPMODE option of DSNCRCCT macro 2-412
 DPMODI option of DSNCRCCT macro 2-400
 DPROP SUPPORT
 field of panel DSNTIPO 2-162
 DPropNR (Data Propagator NonRelational) 1-41
 DRDA (Distributed Relational Database Architecture). 3-4
 DRDA access
 definition 3-3
 organization application 2-355
 setting up 3-3
 specifying modes 3-30
 updating 2-357
 DRDA PORT
 field of panel DSNTIP5 3-5
 DRDA RDBNAM (relational database name)
 See *also* location name
 description 3-11
 DRESPL option of APPL statement 3-17
 DSESLIM option of APPL statement
 CNOS negotiation 3-37
 description 3-15
 DSMAX
 field of panel DSNTIPC 2-216
 DSMAX limit on open data sets
 description 2-45
 DSN command of TSO
 command processor
 description 1-37
 invoking 1-37
 DSN command processor 1-37
 See *also* DSN command of TSO
 DSN1CHKR
 migration preparation 2-272
 DSN1COPY
 migration preparation 2-272
 DSN1DIST address space 2-35
 DSN3@ATH connection exit routine 2-241
 See *also* connection exit routine
 DSN3@SGN sign-on exit routine 2-241
 See *also* sign-on exit routine
 DSN3EPX load module 2-234
 DSN3INI load module 2-234
 DSN6ARVP macro 2-240
 DSN6ENV macro 2-240
 DSN6FAC macro 2-240
 DSN6GRP macro 2-240
 DSN6LOGP macro 2-240
 DSN6SPRM macro 2-240
 DSN6SYSP
 macro 2-240
 DSN8EAE1 exit routine 2-363
 DSN8HUFF edit routine 2-363
 DSNBIND plan name in SYSIBM.MODESELECT table 3-32
 DSNCONNS ISPF table 2-245, 2-277
 DSNCRCCT macro
 description 2-398

DSNCRCT macro (*continued*)
 specification requirements 2-398
 TYPE=COMD 2-406
 TYPE=DSECT 2-417
 TYPE=ENTRY
 description 2-409
 null specification (*) 2-410
 option descriptions 2-409, 2-416
 TYPE=FINAL 2-417
 TYPE=INIT
 description 2-399
 option descriptions 2-400, 2-406
 SUFFIX option 2-396
 TYPE=POOL
 description 2-407
 option descriptions 2-407, 2-408
 DSNDB04 default database 1-22
 See also default database (DSNDB04)
 DSNDB06 database
 installation job DSNTIJIN 2-239
 DSNDBRMS ISPF table 2-245, 2-277
 DSNDDF database 3-3
 See also CDB (communications database)
 DSNEMCO1 CLIST 2-244
 DSNEPRI panel of ISPF 2-249
 DSNETBLS data set for ISPF tables 2-245, 2-277
 DSNHASM precompiler procedure 2-237
 DSNHC precompiler procedure 2-237
 DSNHCOB precompiler procedure 2-237
 DSNHCOB2 precompiler procedure 2-237
 DSNHCPP precompiler procedure 2-237
 DSNHCPP2 precompiler procedure 2-237
 DSNHDECP load module
 installation 2-240
 migration 2-280
 DSNHFOR precompiler procedure 2-237
 DSNHICB2 precompiler procedure 2-237
 DSNHICOB precompiler procedure 2-237
 DSNHPLI precompiler procedure 2-237
 DSNMAPN macro 2-376, 2-378
 DSNPLPKN ISPF table 2-245, 2-277
 DSNTEJ1T job X-63
 DSNTEJAC job 2-71
 DSNTEJAE job 2-71
 DSNTEJAP job 2-71
 DSNTEJRC job 2-71
 DSNTEJxx job
 installation 2-306
 migration 2-12, 2-289
 DSNTEP2 2-368
 DSNTESA job 2-320
 DSNTESC job 2-320
 DSNTESE job 2-320
 DSNTESQ queries 2-274, 2-365
 DSNTIAC subroutine
 calling 2-398
 DSNTIAD sample program
 description 2-367
 executes SQL statements 2-366
 invoked by DSNTEJ1 2-313
 run by DSNTIJTM 2-252, 2-287
 DSNTIAUL sample program 2-368
 DSNTIDxx member 2-104
 DSNTIJAC job 2-71, 2-84
 DSNTIJAE job 2-71, 2-80
 DSNTIJAL job 2-80
 DSNTIJAP job 2-71, 2-83
 DSNTIJCA job 2-238
 DSNTIJDE job 2-238
 DSNTIJEX job
 installation 2-241
 migration 2-284
 DSNTIJFV job 2-301
 DSNTIJIC job
 installation 2-256
 migration 2-274, 2-289
 DSNTIJID job 2-239
 DSNTIJIN job 2-239, 2-284
 DSNTIJMV job 2-233, 2-281
 DSNTIJRC job 2-71, 2-81
 DSNTIJSE job 2-71
 DSNTIJSG job
 installation 2-254
 migration 2-287
 DSNTIJSU job 2-380
 DSNTIJTC job 2-286
 DSNTIJTM job 2-252, 2-287
 DSNTIJUD job 2-71, 2-82
 DSNTIJUZ job 2-240, 2-280
 DSNTIJVC job 2-244, 2-276
 DSNTINS0 CLIST 2-91
 DSNTINS0 CLIST description 2-89, 2-91
 DSNTINST CLIST
 description 2-9
 DSNTNJxx jobs 2-71, 2-72
 DSNUTILB entry in PPT 2-233
 DSNUTILS stored procedure 2-338
 DSNUTILS stored procedure 2-254
 DSNWZP stored procedure 2-254
 DSNYASCP entry in PPT 2-233
 DSNZPARM initialization module 2-223
 See also subsystem parameter module
 dual logging
 description 1-28
 specifying 2-112
 dump
 data set size 2-23
 DXRRML00 entry in PPT 2-233
 dynamic plan selection
 CICS 2-412
 dynamic SQL
 DSNTESA 2-363

dynamic SQL (*continued*)

DSNTESQ 2-363
programs using 2-366

DYNRULS

DSNHDECP parameter 2-226

E

early code 2-74

EAS option of APPL statement 3-17

ECSA (extended common storage area) 2-178

edit routine

DSN8HUFF 2-363

EDM pool

package size 2-41

plan size 2-41

size 2-41

EDMPOOL STORAGE SIZE

field of panel DSNTIPC 2-216

ENCR option of APPL statement 3-17

ENCRYPTPSWDS

column of LUNAMES table 3-22

END

option of DSNMAPN macro

description 2-378

installation format 2-376

environment, operating

DB2 1-35

MVS 1-35

ERLY code 2-74

ERRDEST option

DSNCRCT macro 2-400

EXECUTED STMTS

field of panel DSNTIPD 2-142

EXIT LIBRARY

field of panel DSNTIPT 2-119

exit routine

description 2-363

EXPLAIN PROCESSING

field of panel DSNTIPO

description 2-161

extended common storage area (ECSA) 2-178

extended English code page X-59

extended Katakana code page X-59

extended recovery facility (XRF) 1-39

See also XRF (extended recovery facility)

EXTENDED SECURITY

field of panel DSNTIPR 2-204

external storage 2-15

See also storage

F

fallback

description 2-297

frozen objects 2-298

fallback (*continued*)

jobs 2-301

release incompatibilities 2-299

remigration following 2-303

steps 2-13

FMPROF option of MODEENT macro 3-20

FORMAT

command in sample application 2-326

FORTTRAN COMPILER

field of panel DSNTIPG 2-131

FORTTRAN LINK EDIT

field of panel DSNTIPG 2-131

four 2-252

frozen objects 2-298

function keys 2-344

G

GENERIC

column of LUNAMES table 3-12

description 3-23

global deadlock cycle 2-179

global resource serialization (GRS) 1-39

See also GRS (global resource serialization)

governor (resource limit facility) 2-159

See also resource limit facility (governor)

graphic coded character set identifiers X-58

GROUP ATTACH

field of panel DSNTIPK 2-111

group name

field of panel DSNTIPK 2-110

GROUP option

DSNCRCT macro 2-410

GRS (global resource serialization)

for shared DASD and XRF 1-39

H

HAVAIL option of APPL statement 3-17

help

DB2 UTILITIES panel 1-31

online 2-51, 2-67

Hierarchical Storage Manager (DFSMSHsm) 2-158

See also DFSMSHsm (Hierarchical Storage Manager)

hiperpool

description 1-29

requirements 1-29

storage requirement 2-39

hiperspace

description 1-29

requirements 1-29

Huffman compression

exit routine 2-363

I/O error

- availability of table space after 1-32

IBM COBOL fields of DSNTIPQ 2-128, 2-129

IBMDB2LM mode 3-18

IBMRDB mode 3-18

IBMUSER authority, establishing authorization ID 2-250

IDLE THREAD TIMEOUT

- field of panel DSNTIPR 3-59

IDRC 2-198

IEAAPFxx member of SYS1.PARMLIB

- DSNTIJMV job
 - installation 2-235
 - migration 2-282
- DSNTIPM panel 2-186

IEBCOPY utility 2-73

IEBTPCH utility 2-73

IEFSSNxx member of SYS1.PARMLIB

- DSNTIJMV job
 - installation 2-234
 - migration 2-282
- DSNTIPM panel 2-186

IMS

- attachment facility 2-371
- connecting to DB2
 - controlling 1-40
 - installation 2-250, 2-371
- facilities
 - XRF (extended recovery facility) 1-40
- language interface module (DFSLI000)
 - generating 2-376
- migration 2-278
- operating
 - recovery from system failure 1-40
 - starting 2-326
 - testing 2-323
- programming
 - application 1-41
- system administration 1-42
- using with DB2 1-40

IMS RESLIB

- field of panel DSNTIPW 2-134

IMS.PROCLIB library

- description 2-372

IMSCTRL macro 2-372

incompatibilities of releases 2-290, 2-294

index

- description 1-24
- space
 - description 1-24
 - estimating size 2-29, 2-32
- types
 - clustering 1-25

initial program load (IPL) 2-251

- See *a/so* IPL (initial program load)

INPUT BUFFER

- field of panel DSNTIPL 2-190

INPUT MEMBER NAME

- field of panel DSNTIPA1 2-103

INSTALL DD CONTROL SUPT.

- field of panel DSNTIPZ 2-209

INSTALL IRLM

- field of panel DSNTIPI 2-174

INSTALL TYPE

- field of panel DSNTIPA1 2-99

installation

- character conversion connections X-59
- CLIST 2-9, 2-89
- defining DB2 to MVS 2-233
- description 2-10
- jobs
 - description 2-9
 - DSNTIJCA 2-238
 - DSNTIJDE 2-238
 - DSNTIJEX 2-241
 - DSNTIJIC 2-256
 - DSNTIJID 2-239
 - DSNTIJIN 2-239
 - DSNTIJMV 2-233, 2-281
 - DSNTIJSG 2-254
 - DSNTIJSU 2-380
 - DSNTIJTM 2-252, 2-287
 - DSNTIJUZ 2-240
 - DSNTIJVC 2-244
 - system affinity 2-212
- macros
 - DSN6ARVP 2-240
 - DSN6ENV 2-240
 - DSN6FAC 2-240
 - DSN6GRP 2-240
 - DSN6LOGP 2-240
 - DSN6SPRM 2-240
 - DSN6SYSP 2-240
- output 2-90
- overview 2-257
- panels
 - description 2-91
 - list of 2-92
- planning 2-9
- second subsystem 2-257
- steps 2-11
- tailoring session 2-10
- verification
 - planning 2-308
 - testing batch environment 2-316
 - testing CICS environment 2-327
 - testing IMS environment 2-323
 - testing PL/I batch 2-319
 - testing SPUFI 2-319
 - testing the TSO attachment facility 2-313

- Installation CLIST
 - calculating DASD requirements 2-24
 - installation SYSADM authority
 - authorization IDs established 2-250
 - installation SYSOPR authority
 - authorization IDs established 2-250
 - installation tapes or cartridges 2-69
 - installation verification procedure (IVP) 2-303
 - See *also* IVP (installation verification procedure)
 - integrated catalog facility
 - DSNTIJCA job 2-238
 - Interactive System Productivity Facility (ISPF) 1-31
 - See *also* ISPF (Interactive System Productivity Facility)
 - invoking
 - CLIST 2-90
 - DSN command processor 1-37
 - IOBUF buffer pool
 - calculating storage requirements 3-36
 - description 3-25
 - IP address
 - definition 3-49
 - IPADDR
 - column of IPNAMES table 3-57
 - IPL (initial program load)
 - installation 2-251
 - migration 2-285
 - IRLM (internal resource lock manager)
 - address space 2-35
 - dump formatting module, AMDPRECT 2-234
 - entry in PPT 2-233
 - starting
 - after fallback 2-302
 - installation 2-251
 - migration 2-285
 - IRLM XCF GROUP NAME
 - field of panel DSNTIPJ 2-180
 - IS SAMPLES GROUP NEW?
 - field of panel DSNTIP3 2-137
 - ISPF (Interactive System Productivity Facility)
 - DB2 considerations 1-37
 - primary option menu
 - panel connection 2-246
 - requirement 1-42
 - system administration 1-37
 - tutorial panels 1-31
 - ISPF ISPLINK MODULE
 - field of panel DSNTIPW 2-134
 - ISPF Skeleton Library (ISPSLIB) 2-89
 - ISTINCLM mode table 3-15, 3-18
 - IVP (installation verification procedure)
 - COBOL options 2-308
 - fallback 2-303
 - installation 2-305
 - migration 2-305
 - phase 2-331
 - IVP (installation verification procedure) (*continued*)
 - phases 2-312
 - PL/I options 2-312
 - programs-phases relationship 2-306
- ## J
- JOB statement
 - description 2-74
- ## L
- LANGUAGE DEFAULT
 - field of panel DSNTIPF 2-165
 - language interface token (LIT) 2-372
 - LE/370 RUN TIME 3-52
 - field of panel DSNTIPG 2-132
 - library
 - description 2-69
 - distribution 2-70
 - online 1-7
 - prefix name 2-78
 - target 2-70
 - link checker 2-272
 - See *also* DSN1CHKR
 - LINK LIST ENTRY
 - field of panel DSNTIPM 2-189
 - LINK LIST LIBRARY
 - field of panel DSNTIPT 2-119
 - LINK LIST SEQUENCE
 - field of panel DSNTIPM 2-189
 - LINKNAME
 - column of IPNAMES table 3-57
 - column of LOCATIONS table 3-5, 3-21, 3-56
 - column of USERNAMES table 3-23, 3-58
 - LIST NAME
 - field of panel DSNTIP3 2-137
 - LIT (language interface token) 2-372
 - LLA (LNKLST lookaside)
 - description 2-74
 - installation 2-236
 - migration 2-282
 - LMDENT option of APPL statement 3-17
 - LNKLST lookaside (LLA) 2-74
 - See *also* LLA (LNKLST lookaside)
 - LNKLSTxx member of SYS1.PARMLIB
 - installation 2-186, 2-236
 - migration 2-282
 - LOAD DISTRIBUTION
 - field of panel DSNTIPT 2-119
 - LOAD LIBRARY
 - field of panel DSNTIPT 2-119
 - load module library
 - DSNHDECP 2-240, 2-280
 - SDSNEXIT 2-75
 - SDSNLINK 2-74

load module library (*continued*)
 SDSNLOAD 2-75

loading 2-69
 See also LOAD utility
 distribution libraries 2-69
 target libraries 2-69

LOCAL DATE LENGTH
 field of panel DSNTIP4 2-171

local deadlock cycles 2-179

LOCAL TIME LENGTH
 field of panel DSNTIP4 2-171

LOCATION
 column of LOCATIONS table 3-21, 3-56

location name
 description 3-11
 updating the BSDS 3-46

lock
 object
 request 2-175

LOCK ENTRY SIZE
 field of panel DSNTIPJ 2-180

LOCKS PER TABLE(SPACE)
 field of panel DSNTIPJ 2-178

LOCKS PER USER
 field of panel DSNTIPJ 2-178

log
 data set
 shared read-only data 2-263
 storage examples 2-19

log mode table 3-18
 See also mode table

LOG PASSWORD
 field of panel DSNTIPP 2-182

log range directory 1-27

logical unit (LU) 3-9

LOGMODE option of MODEENT macro 3-19

logon mode table entries 3-18

LU 6.2 communications protocols 3-4

LUNAME 3-4
 column of LUMODES table 3-31
 column of LUNAMES table 3-22
 column of MODESELECT table 3-32
 field of panel DSNTIPR 3-5
 option of APPL statement
 coding values 3-14
 description 3-12
 updating the BSDS 3-46

M

MACRO LIBRARY
 field of panel DSNTIPT 2-119

main storage 2-38
 See also storage

MAX ABEND COUNT
 field of panel DSNTIPX 2-208

MAX BATCH CONNECT
 field of panel DSNTIPE 2-147

MAX KEPT DYN STMTS
 field of panel DSNTIPE 2-148

MAX REMOTE ACTIVE
 field of panel DSNTIPE 2-146

MAX REMOTE CONNECTED
 field of panel DSNTIPE 2-147

MAX TSO CONNECT
 field of panel DSNTIPE 2-147

MAX USERS
 field of panel DSNTIPE 2-146

MAXBFRU option of LINE statement 3-42

MAXDATA option of VTAM
 considerations for NCP connections the 3-43

MAXIMUM ECSA
 field of panel DSNTIPJ 2-178

MAXPVT option of APPL statement 3-17

MEMBER IDENTIFIER
 field of panel DSNTIPJ 2-179

MEMBER NAME
 field of panel DSNTIPK
 specifying 2-110

migration
 CLIST 2-9, 2-89
 considerations 2-267
 defining 2-281
 jobs
 description 2-9
 DSNTIJEX 2-284
 DSNTIJIC 2-274
 DSNTIJIN 2-284
 DSNTIJMV 2-281
 DSNTIJTC 2-286
 DSNTIJUZ 2-280
 DSNTIJVC 2-276
 planning 2-9
 release incompatibilities 2-290
 steps 2-12
 tailoring session 2-10

MINIMUM DIVIDE SCALE
 field of panel DSNTIPF 2-166

mixed applications, calculating session limits for 3-34

mixed CCSIDs (coded character set identifiers) X-58

MIXED DATA
 field of panel DSNTIPF 2-167

MNOTE macro error 2-378

mode of VTAM
 adding new modes 3-20
 creating new modes 3-20
 VTAM sessions
 associating with sessions 3-29
 default 3-18
 IBMDB2LM 3-18
 IBMRDB 3-18
 SNASVCMG 3-18

- mode table
 - default 3-15, 3-18
 - entering modes 3-18
- MODEENT macro 3-18, 3-20
- MODENAME column
 - LUMODES table 3-31
 - MODESELECT table 3-32
- MODESELECT
 - column of LUNAMES table 3-22
- MODETAB option of APPL statement 3-15
- modified jobs 2-91
- MODIFY VTAM,DEFINE command of VTAM 3-28
- MONITOR SIZE
 - field of panel DSNTIPN 2-156
- MONITOR TRACE
 - field of panel DSNTIPN 2-156
- monitoring
 - VTAM buffer pools
 - DISPLAY NET, BFRUSE 3-25
 - MODIFY command of MVS 3-25
- multi-site update
 - APPL options 3-16
 - description 3-3
 - MODEENT options 3-19
- MVS
 - DB2 considerations 1-33
 - defining DB2 to 2-233
 - environment 1-33
 - IPL 2-251
- MVS CATALOG
 - field of panel DSNTIPP 2-182
- MVS PARMLIB updates 2-186

N

- national language
 - character set identifiers X-56
 - double-byte character set identifiers X-58
- NCP-connected DB2s
 - considerations for MAXDATA option 3-43
 - sample definitions 3-43
- NETID 3-4, 3-11
- NetView
 - installation 2-257
- NEWAUTHID
 - column of USERNAMES table 3-23, 3-58
- NUMBER OF COPIES
 - field of panel DSNTIPH 2-113, 2-114
- NUMBER OF LOGS
 - field of panel DSNTIPL 2-190
- NUMBER OF TCBS
 - field of panel DSNTIPX 2-208

O

- OBJT REGISTRATION TABLE
 - field of panel DSNTIPZ 2-211
- online books 1-7
- OpenEdition 3-52, 3-54, 3-58
- operation
 - continuous 1-32
- operator
 - CICS 1-40
 - not required for IMS start 1-40
 - START command 1-37
- OPERCNOS option of APPL statement 3-17
- OPTION
 - option of DSNMAPN macro 2-376
- organization application
 - DRDA access 2-355
 - format command 2-326
 - panels 2-341
 - with IMS 2-326
- OUTPUT BUFFER
 - field of panel DSNTIPL 2-191
- OUTPUT MEMBER NAME
 - field of panel DSNTIPA1 2-103
- overriding built-in conversion X-62

P

- pacing
 - controlling
 - description 3-26
 - modifying session limits 3-28
 - SRCVPAC option of MODEENT macro 3-20, 3-28
 - VPACING option of VTAM APPL statement 3-16
- PACKAGE LISTS
 - field of panel DSNTIPD 2-142
- PACKAGE STATEMENTS
 - field of panel DSNTIPD 2-142
- PACKAGES
 - field of panel DSNTIPD 2-141
- page
 - description 1-22
- panel
 - directory 2-92
 - ISPF 2-10, 2-89
 - organization 2-341
 - projects 2-341
- PARAMETER MODULE
 - field of panel DSNTIPO
 - description 2-160
- PARMLIB update options 2-186
- PARSESS option of APPL statement 3-17
- partition
 - reorganizing 1-32

- partitioned data set, managing 1-36
- partitioned index
 - description 1-25
- partitioned table space
 - description 1-23
- password
 - column of USERNAMES table 3-24, 3-58
 - VTAM 3-12
 - See also VTAM (Virtual Telecommunications Access Method)
- PCT (program control table) 2-389
- PCTEROP option of DSNCRCT macro 2-400
- performance
 - improving X-59
- PERMANENT UNIT NAME
 - field of panel DSNTIPA2 2-109
- phone application 2-351
 - data set format 2-353
 - format command 2-326
 - JCL 2-354
 - panels 2-352
 - program description 2-351
 - transaction code 2-330
 - using under batch 2-353
- PIU (path information unit)
 - description 3-36
 - relationship to MAXBFRU 3-42
- PL/I fields of panel DSNTIPG 2-131, 2-132
- PLAN
 - option of DSNCRCT macro 2-412
 - option of DSNMAPN macro
 - description 2-378
 - installation format 2-376
- PLAN AUTH CACHE
 - field of DSNTIPP 2-185
- plan name for remote bind 3-32
- plan size, calculating 2-42
- PLAN STATEMENTS
 - field of panel DSNTIPD 2-141
- PLANI option of DSNCRCT macro 2-401
- PLANNAME column
 - SYSIBM.MODESELECT catalog table 3-31
- PLANS
 - field of panel DSNTIPD 2-141
- PLNEXIT option of DSNCRCT macro 2-412
- PLNPGME option of DSNCRCT macro 2-412
- PLNPGMI option of DSNCRCT macro 2-401
- PLNXTR1 option of DSNCRCT macro 2-401
- PLNXTR2 option of DSNCRCT macro 2-401
- PLT (program list table) processing 2-394
- populating 3-20, 3-56
 - CDB
 - connecting subsystems 3-20, 3-21, 3-56, 3-57
 - installation 2-256
- port
 - column of LOCATIONS table 3-56

- port (*continued*)
 - definition 3-49
 - ephemeral 3-49
 - server 3-49
 - well-known 3-49
- port numbers 3-53
- PPT (CICS processing program table) 2-393
- PPT (MVS program properties table) 2-233
- prefix
 - active log 2-114
 - archive log 2-115, 2-194
 - field of panel DSNTIPA1 2-102
 - library 2-78
 - log 2-115
- PRIMARY QUANTITY
 - field of panel DSNTIPA 2-193
- PRIPROT option of MODEENT macro 3-20
- PROC NAME
 - field of panel DSNTIPI 2-176
- processing program table (PPT) 2-393
 - See also PPT (CICS processing program table)
- program control table (PCT) 2-389
- program libraries 2-74
- program list table (PLT) 2-394
- PROGxx member of SYS1.PARMLIB 2-235, 2-282
- project application
 - format command 2-326
 - panels 2-341
 - using 2-345
 - with IMS 2-326
- PRTCT option of APPL statement 3-12, 3-15
- PSERVIC option of MODEENT macro 3-20
- PSTOP transaction type 2-378
- PURGEC option of DSNCRCT macro
 - description 2-402

Q

- QUIESCE PERIOD
 - field of panel DSNTIPA 2-197

R

- RACF (Resource Access Control Facility)
 - DB2 considerations 1-36
 - option to specify at installation or migration 2-183
- RCT (resource control table)
 - defining DB2 to CICS 2-385
 - description 2-398
 - installation 2-250
- RDO (resource definition online) 2-387, 2-390
- READ TAPE UNITS
 - field of panel DSNTIPA 2-195
- reading
 - online book 2-65

real storage 2-48
 See also storage
 rebinding
 remote package X-64
 RECALL DATABASE
 field of panel DSNTIPO 2-158
 RECALL DELAY
 field of panel DSNTIPO 2-159
 RECEIVE job 2-81
 record
 description 1-24
 RECORDING MAX
 field of panel DSNTIPA
 description 2-196
 recovery job DSNTIJDE 2-238
 recovery log
 description 1-28
 REGISTRATION DATABASE
 field of panel DSNTIPZ 2-211
 REGISTRATION OWNER
 field of panel DSNTIPZ 2-210
 relational database 1-19
 relational database name 3-11
 release incompatibilities 2-290
 remigration 2-14, 2-303
 REMOTE LOCATION
 field of panel DSNTIPY 2-213
 remote unit of work 3-3
 See also DRDA access
 REO (region error option)
 default OPTION of DSNMAPN macro 2-378
 installation 2-373
 REQUIRE FULL NAMES
 field of panel DSNTIPZ 2-210
 Resource Access Control Facility (RACF) 2-183
 See also RACF (Resource Access Control Facility)
 RESOURCE AUTHID
 field of panel DSNTIPP 2-184
 resource control table (RCT) 2-250, 2-385
 See also RCT (resource control table)
 resource definition online (RDO) 2-387
 resource limit facility (governor) 2-159
 authorization ID 2-184
 creating database
 during installation 2-255
 database 1-30
 specifying 2-159
 storage estimation 2-24
 RESOURCE TIMEOUT
 field of panel DSNTIPI 2-175
 resource translation table (RTT) 2-372
 See also RTT (resource translation table)
 RESTART
 field of panel DSNTIPS 2-200
 RESYNC PORT
 field of panel DSNTIP5 3-5, 3-51
 RETENTION PERIOD
 field of panel DSNTIPA 2-197
 RID (record identifier) pool
 size 2-41
 RID blocks 2-41
 RID POOL SIZE
 field of panel DSNTIPC 2-217
 RLF AUTO START
 field of panel DSNTIPO 2-159
 RLST ACCESS ERROR field
 panel DSNTIPO 2-159
 panel DSNTIPR 2-202
 RLST NAME SUFFIX
 field of panel DSNTIPO 2-159
 ROLBE option of DSNCRCT macro 2-412
 ROLBI option of DSNCRCT macro 2-402
 rollback
 ROLBI option of DSNCRCT macro 2-402
 sync point rollback 2-402
 row
 description 1-19
 RTT (resource translation table)
 description 2-377
 installation 2-372
 RUNLIB.LOAD library
 DASD volume 2-108
 device type 2-109
 DSNTIJIN job 2-239
 installing a second DB2 2-260, 2-262
 naming considerations 2-78
 RUSIZES option of MODEENT macro 3-20

S

sample application
 description 2-341
 for verifying installation 2-305
 IMS ODBA stored procedure 2-339
 organization 2-346
 output X-3
 phone 2-351
 project 2-345
 unload program (DSNTIAUL) 2-368
 utilities stored procedure 2-338
 SAMPLE LIBRARY
 field of panel DSNTIPT 2-118
 sample VTAM definitions 3-43
 description 3-39
 NCP-connected DB2s 3-43
 SCT02 table space
 description 1-27
 SDSNCHDR library 2-71
 SDSNCLST library 2-71
 SDSNDBRM library 2-71
 SDSNENU library 2-71

SDSNEXIT library 2-71, 2-240
 SDSNLINK library
 description 2-71, 2-74
 suffix 2-189
 SDSNLOAD library
 description 2-71
 link list options 2-74
 SDSNMACS library 2-71
 SDSNSAMP library
 description 2-71
 output from panel session 2-90
 source for DSNTIAUL 2-368
 SDSNSPFM library 2-71
 SDSNSPFP library 2-71
 SDSNSPFPE library 2-71
 SDSNSPFPK library 2-71
 SDSNSPFS library 2-71
 SDSNSPFT library 2-71
 SDXRRESL library 2-71
 SDXRSAMP library 2-71
 SECACPT option of APPL statement 3-15
 SECONDARY QTY
 field of panel DSNTIPA 2-194
 secondary server 3-34
 SECPROT option of MODEENT macro 3-20
 security
 installation 2-243
 migration 2-281
 SECURITY_IN
 column of LUNAMES table 3-22
 SECURITY_OUT
 column of IPNAMES table 3-57
 column of LUNAMES table 3-22
 SELECT statement
 example
 WHERE clause 1-20
 selection of data values 2-343
 SEQUENTIAL CACHE
 field of panel DSNTIPE 2-148
 server
 DB2 as secondary server 3-34
 service name
 definition 3-50
 session 3-9
 session limit for VTAM
 calculating 3-33
 considerations for mixed applications 3-34
 modifying default 3-28
 modifying LUMODES 3-30
 OPERCNOS option of APPL statement 3-17
 specifying in APPL statement 3-15
 SHARE PREPARED STMTS
 field of panel DSNTIP4 2-173
 shared read-only data
 log data set considerations 2-263
 SHDDEST option of DSNCRCCT macro 2-402
 sign-on exit routine
 installation 2-241
 migration 2-284
 SIGNID option of DSNCRCCT macro 2-402, 2-410
 single logging 1-28
 single-byte character set identifiers X-56, X-58
 SIT (system initialization table) 2-386
 SITE TYPE
 field of panel DSNTIPO 2-163
 SKCT (skeleton cursor table)
 description 1-27
 skeleton cursor table (SKCT) 1-27
 See *also* SKCT (skeleton cursor table)
 skeleton package table (SKPT) 1-27
 See *also* SKPT (skeleton package table)
 SKPT (skeleton package table)
 description 1-27
 SMF (System Management Facility)
 buffers 2-243
 installation 2-242
 SMF ACCOUNTING
 field of panel DSNTIPN 2-155
 SMF STATISTICS
 field of panel DSNTIPN 2-156
 SMP/E (System Modification Program/Extended) 2-79
 ACCEPT job 2-71, 2-84
 allocation job 2-80
 APPLY job 2-71, 2-83
 cleanup job 2-71, 2-82
 copying jobs to DASD 2-72
 data set options 2-79
 data sets for two releases 2-80
 job listings 2-71
 loading DB2 libraries 2-69
 RECEIVE job 2-71, 2-81
 RELFILE format 2-69
 sharing data sets with IMS 2-79
 steps 2-11
 SNA sense code X'800A' 3-42, 3-43
 snap dump 2-403
 SNAP option of DSNCRCCT macro 2-403
 SNASVCMG mode 3-18
 softcopy publications 1-7
 SONSCIP option of APPL statement 3-17
 sort
 program
 APF authorization of library 2-236
 SORT LIBRARY
 field of panel DSNTIPW 2-134
 SORT POOL SIZE
 field of panel DSNTIPC 2-217
 SPT01 table space 1-27
 SPUFI
 access to remote systems 2-320
 binding
 migration 2-287

SPUFI (*continued*)
 binding (*continued*)
 remotely 2-320
 testing 2-319
 SQL
 entry level conformance 2-269
 SQL (Structured Query Language)
 description 1-20
 processing conversations
 description 3-30
 specifying mode 3-31
 SQL STRING DELIMITER
 field of panel DSNTIPF 2-166
 SQL/DS support 3-4
 SRBEXIT option of APPL statement 3-16
 SRCLIB.DATA library
 DASD volume 2-108
 device type 2-109
 DSNTIJIN job 2-239
 installing a second DB2 2-260, 2-262
 naming considerations 2-78
 SRCVPAC option of MODEENT macro
 description 3-20
 used to control pacing 3-28
 SSM (subsystem member)
 entry in IMS.PROCLIB 2-372
 execution parameter 2-374
 SSNDPAC option of MODEENT macro
 description 3-20
 recommended value 3-27
 STANDBY option of DSNCRCT macro 2-403
 start options for VTAM 3-24
 starting
 DB2
 installation 2-251
 migration 2-285, 2-302
 IRLM
 after fallback 2-302
 installation 2-251
 migration 2-285
 TSO
 after fallback 2-303
 installation 2-252
 migration 2-286
 statistics
 report destination 2-402
 STATISTICS TIME
 field of panel DSNTIPN 2-156
 STD SQL LANGUAGE
 field of panel DSNTIP4 2-172
 STEPLIB
 statement of DB2 program libraries 2-76
 storage
 calculating
 external storage 2-15
 main 2-38, 2-47
 predefined models 2-15
 storage (*continued*)
 calculating (*continued*)
 real 2-48
 virtual constraints 2-47
 VTAM IOBUF 3-36
 working 2-46
 storage group
 DB2
 description 1-22
 storage management subsystem 1-36
 See also DFSMS (Data Facility Storage Management Subsystem)
 stored procedures
 enabling 2-263
 running concurrently 2-37
 subsystem parameter 2-287
 utilities 2-254, 2-287, 2-338
 Visual Explain 2-254
 STRING DELIMITER
 field of panel DSNTIPF 2-166
 STRTWT option of DSNCRCT macro 2-403
 structure, description 1-19
 Structured Query Language (SQL) 1-20
 See also SQL (Structured Query Language)
 SUBID option of DSNCRCT macro 2-404
 subsystem
 multiple 2-76, 2-257
 name
 IEFSSNxx member of SYS1.PARMLIB 2-234
 subsystem command prefix 1-36
 SUBSYSTEM MEMBER
 field of panel DSNTIPM 2-188
 SUBSYSTEM NAME field
 panel DSNTIPI 2-175
 panel DSNTIPM 2-186
 subsystem parameter
 ABIND 2-226
 EDMBFIT 2-227
 OJPERFEH 2-227
 PARAMDEG 2-227
 PARTKEYU 2-227
 PTASKROL 2-227
 RETVLCFK 2-227
 XLKUPDLT 2-227
 subsystem parameter module
 directory 2-223
 installation 2-240
 migration 2-280
 option descriptions 2-113, 2-216
 SUBSYSTEM SEQUENCE
 field of panel DSNTIPM 2-188
 subtask priority 2-400, 2-412
 SUFFIX
 field of panel DSNTIPA1 2-102
 option of DSNCRCT macro 2-404

- suffix identification character 2-404
- sync point rollback 2-402
- SYNCLVL option of APPL statement 3-16
- syntax diagrams, how to read 1-4
- SYS1.PARMLIB library
 - updated by DSNTIJMV 2-233, 2-281
- SYS1.PROCLIB library 2-233, 2-281
- SYS1.VTAMLST library 3-24
- SYSADM authority
 - establish authorization IDs 2-250
 - use 2-257
- SYSIBM.IPNAMES table of CDB
 - description 3-57
- SYSIBM.LOCATIONS table of CDB
 - description 3-21, 3-56
 - example 3-21, 3-57
- SYSIBM.LUMODES table of CDB
 - CONVLIMIT column 3-31
 - description 3-30
 - LUNAME column 3-31
 - MODENAME column 3-31
 - updating 3-30
- SYSIBM.LUNAMES table of CDB
 - description 3-22
 - example 3-23
 - specifying modes 3-30
 - updating 3-30
- SYSIBM.MODESELECT table of CDB
 - bind plan name 3-32
 - description 3-31
 - search order 3-32
- SYSIBM.USERNAMES table of CDB
 - description 3-23, 3-58
- SYSLGRNX directory table
 - table space
 - description 1-27
- SYSOPR authority
 - establish authorization IDs 2-250
- system 2-18
 - conversation
 - specifying modes 3-29
 - using default mode 3-22
 - structures 1-26, 1-30
 - utilities directory 1-27
- SYSTEM ADMIN 1
 - field of panel DSNTIPP 2-183
- SYSTEM ADMIN 2
 - field of panel DSNTIPP 2-183
- system initialization table (SIT) 2-386
- SYSTEM MACLIB
 - field of panel DSNTIPW 2-133
- System Management Facility (SMF) 2-242
 - See also SMF (System Management Facility)
- System Modification Program/Extended (SMP/E) 2-69
 - See also SMP/E (System Modification Program/Extended)

- SYSTEM OPERATOR 1
 - field of panel DSNTIPP
 - description 2-184
- SYSTEM OPERATOR 2
 - field of panel DSNTIPP
 - description 2-184
- SYSTEM PROCEDURES
 - field of panel DSNTIPW 2-133
- system services address space 2-35
- SYSUTILX table space 1-27

T

- table
 - CICS table generation procedure 2-398
 - description 1-24
 - estimating storage 2-27
 - relationship to views 1-25
- table space
 - description 1-22
 - reorganizing
 - separately from partitions 1-32
- TABLE SPACES
 - field of panel DSNTIPD 2-141
- TABLES
 - field of panel DSNTIPD 2-140
- TABLES IN STMT
 - field of panel DSNTIPD 2-142
- target library 2-70
- TASKREQ option of DSNCRCT macro 2-413
- TCPALVER
 - field of panel DSNTIP5 2-206, 3-51
- TEMP 32K DATA SETS
 - field of panel DSNTIPD 2-144
- TEMP 32K SPACE
 - field of panel DSNTIPD 2-143, 2-252
- TEMP 4K DATA SETS
 - field of panel DSNTIPD 2-143
- TEMP 4K SPACE
 - field of panel DSNTIPD 2-142, 2-252
- TEMP CLIST LIBRARY
 - field of panel DSNTIPT 2-117
- TEMP data set 2-90
- temporary
 - database storage estimation 2-21, 2-142
- TEMPORARY UNIT NAME
 - field of panel DSNTIPA2 2-109
- TERM option
 - DSNCRCT macro 2-411
- thirty-two 2-252
- THRDA option
 - DSNCRCT macro 2-413
- THRDM option of DSNCRCT macro 2-414
- THRDMAX option of DSNCRCT macro 2-404
- THRDS option of DSNCRCT macro 2-414

thread
 CICS
 maximum, specifying 2-404, 2-406
 maximum number 2-404
 space 2-47
 subtasks
 priority 2-400, 2-412
 termination
 disconnecting 2-408
 DSNCRCT macro 2-408
 TIME FORMAT
 field of panel DSNTIP4
 description 2-171
 TIME TO AUTOSTART
 field of panel DSNTIPI 2-176
 TIMEOUT VALUE
 field of panel DSNTIPX 2-208
 TIMESTAMP ARCHIVES
 field of panel DSNTIPH 2-115
 TOKENE option of DSNCRCT macro 2-414
 TOKENI option of DSNCRCT macro 2-405
 TPN
 column of LOCATIONS table 3-21
 TPN (transaction program name) 3-13
 trace
 identification for CICS 2-405
 TRACE AUTO START
 field of panel DSNTIPN 2-155
 TRACE SIZE
 field of panel DSNTIPN 2-155
 TRACEID option of DSNCRCT macro 2-405
 transaction entries 2-387
 transaction program name (TPN)
 description 3-13
 transmitting character data X-55
 TSO
 application programs
 batch 1-36
 foreground 1-36
 DB2 considerations 1-36
 establishing user IDs 2-250
 installation procedures 2-243
 migration procedures 2-275
 requirement 1-42
 starting 2-252
 testing 2-313
 TSPROF option of MODEENT macro
 description 3-20
 tuning
 VTAM
 buffer storage 3-25
 creating new modes 3-20
 description 3-24
 MODEENT macro 3-27
 pacing count 3-26
 session limits 3-28
 SRCVPAC option of MODEENT macro 3-28
 TWAIT option of DSNCRCT macro
 TYPE=ENTRY macro 2-415
 TWAITI option of DSNCRCT macro 2-406
 two-phase commit 3-4, 3-16, 3-51
 DRDA (Distributed Relational Database
 Architecture) 3-4
 TXID option of DSNCRCT macro 2-411, 2-416
 TXIDSO option of DSNCRCT macro
 description 2-405
 TYPE
 column of USERNAMES table 3-23, 3-58
 option of MODEENT macro 3-20
 TYPE=COMD option of DSNCRCT macro 2-406
 TYPE=ENTRY option of DSNCRCT macro 2-409
 TYPE=FINAL option of DSNCRCT macro 2-417
 TYPE=INIT option of DSNCRCT macro 2-399
 TYPE=POOL option of DSNCRCT macro 2-407

U

U LOCK FOR RR/RS
 field of panel DSNTIPI 2-176
 UNKNOWN AUTHID
 field of panel DSNTIPP 2-184
 unload 2-368
 See *also* DSNTIAUL sample program
 UNREGISTERED DDL DEFAULT
 field of panel DSNTIPZ 2-210
 UPDATE
 multi-site 3-3
 See *also* multi-site update
 updating
 activity 2-345
 communications database while DDF is active 3-32
 LUMODES table of CDB 3-30
 LUNAMES table of CDB 3-30
 MODESELECT table of CDB 3-31
 session limits 3-30
 system parameters 2-228
 UR CHECK FREQ
 field of panel DSNTIPN 2-157
 USE PROTECTION
 field of panel DSNTIPP 2-183
 USER
 option of DSNCRCT macro 2-411
 user-defined characters X-58
 USERID option of DSNCRCT macro 2-411
 USERNAMES
 column of IPNAMES table 3-57
 column of LUNAMES table 3-23
 utilities
 description 1-31
 types
 CATMAINT 2-286
 UTILITY CACHE OPTION
 field of panel DSNTIPE 2-148

working storage 2-46
 See also storage
WRITE THRESHOLD
 field of panel DSNTIPL 2-191
WRITE TO OPER
 field of panel DSNTIPA 2-196
WTO ROUTE CODES
 field of panel DSNTIPO 2-158
WTOR ROUTE CODE
 field of panel DSNTIPA 2-196

X

XRF (extended recovery facility)
 CICS toleration 1-39
 IMS toleration 1-40

Z

ZPARAM 2-223
 See also subsystem parameter module

We'd Like to Hear from You

DB2 for OS/390
Version 5
Installation Guide
Publication No. GC26-8970-03

Please use one of the following ways to send us your comments about this book:

- Mail—Use the Readers' Comments form on the next page. If you are sending the form from a country other than the United States, give it to your local IBM branch office or IBM representative for mailing.
- Fax—Use the Readers' Comments form on the next page and fax it to this U.S. number: 800-426-7773 or (408) 463-4393.
- Electronic mail—Use one of the following network IDs:
 - IBMMail: USIBMXFC @ IBMMAIL
 - IBMLink: DB2PUBS @ STLVM27
 - Internet: DB2PUBS@VNET.IBM.COM

Be sure to include the following with your comments:

- Title and publication number of this book
- Your name, address, and telephone number or your name and electronic address if you would like a reply

Your comments should pertain only to the information in this book and the way the information is presented. To request additional publications, or to comment on other IBM information or the function of IBM products, please give your comments to your IBM representative or to your IBM authorized remarketer.

IBM may use or distribute your comments without obligation.

Readers' Comments

DB2 for OS/390

Version 5

Installation Guide

Publication No. GC26-8970-03

How satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Technically accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Grammatically correct and consistent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Graphically well designed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

May we contact you to discuss your comments? Yes No

Name

Address

Company or Organization

Phone No.

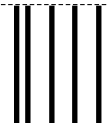


Cut or Fold
Along Line

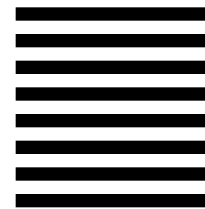
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department BWE/H3
PO Box 49023
San Jose, CA 95161-9945



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Program Number: 5655-DB2



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC26-8970-03

